Perancangan Data Warehouse untuk Analisis Efisiensi dan Kapasitas Pembangkit Listrik Global

Disusun Oleh:

Widya Louisa (23031554180)

Melinda Venta Lydia Siburian (23031554124)

Khalila Salma Dzakia Agram (23031554126)

Dosen Pengampu:

Harmon Prayogi, M. Sc.

Universitas Negeri Surabaya

Surabaya, 2025

DAFTAR ISI

BAB IV : HASIL DAN PEMBAHASAN

BAB I

PENDAHUlUAN

1.1 Latar Belakang

Transformasi digital dan kemajuan teknologi informasi telah mendorong organisasi, baik publik maupun swasta, untuk mengelola data dalam skala besar dan beragam. Salah satu bentuk pengelolaan data strategis yang kini menjadi pondasi dalam proses pengambilan keputusan adalah penerapan data warehouse. Berbeda dari sistem basis data operasional yang bersifat transaksional (OLTP), data warehouse dirancang khusus untuk analisis historis dan agregasi data dalam bentuk multidimensi melalui pendekatan OLAP (*Online Analytical Processing*) (Vaisman & Zimányi, 2014).

Kebutuhan akan sistem *data warehouse* semakin relevan dalam konteks industri energi, khususnya dalam pengelolaan data pembangkit listrik berskala global (*global power plant*). Saat ini, ribuan pembangkit listrik tersebar di seluruh dunia dengan keragaman tipe sumber energi seperti batubara, gas alam, tenaga air, nuklir, serta energi terbarukan seperti angin dan surya. Kompleksitas ini menuntut adanya sistem informasi yang mampu mengintegrasikan data teknis, operasional, dan lingkungan secara menyeluruh. *Data warehouse* memungkinkan pengelolaan dan analisis data penting seperti kapasitas terpasang (*installed capacity*), output energi, efisiensi bahan bakar, tingkat emisi karbon, serta status dan umur operasional unit pembangkit. Hal ini menjadi sangat penting dalam mendukung transparansi, pelaporan kinerja, perencanaan ekspansi energi, dan kebijakan transisi menuju energi bersih dan berkelanjutan.

1.2 Tujuan

Tujuan utama dari penelitian ini adalah untuk merancang, memodelkan, dan mengevaluasi sebuah sistem *data warehouse* yang mampu mengintegrasikan data pembangkit listrik dari berbagai sumber ke dalam satu arsitektur yang terstruktur dan konsisten. Sistem ini dirancang agar dapat mendukung analisis multidimensi melalui penerapan *OLAP cube*, sehingga memungkinkan eksplorasi data yang lebih mendalam dan berbasis histori. Selain itu, penelitian ini juga bertujuan untuk mengimplementasikan teknologi komputasi terdistribusi

untuk meningkatkan efisiensi dalam pengelolaan pipeline data. Dalam hal ini, *Celery* digunakan sebagai pengatur tugas terjadwal (*task scheduler*), sedangkan *RabbitMQ* berperan sebagai *message broker* untuk mendukung komunikasi antar proses secara asinkron.

1.3 Manfaat

Manfaat yang diharapkan dari penelitian ini antara lain adalah: (1) Meningkatkan kemampuan deteksi pola dan anomali dalam data pembangkit listrik yang tersebar secara geografis, yang dapat digunakan sebagai dasar untuk perbaikan operasional atau peringatan dini terhadap potensi masalah.; (2) Menyediakan *framework* referensi yang dapat digunakan oleh instansi pemerintah, lembaga riset, maupun perusahaan energi dalam membangun sistem intelijen energi berbasis *data warehouse* dan OLAP.; dan (3) Mempermudah proses pemantauan dan evaluasi pembangkit listrik, termasuk analisis performa, efisiensi, dan emisi, melalui eksplorasi data secara interaktif dan visual yang dapat disesuaikan dengan kebutuhan pengguna.

BAB II

DASAR TEORI

2.1 Pengertian Data Warehouse

Data warehouse adalah sistem penyimpanan data yang dirancang secara khusus untuk mendukung proses pengambilan keputusan di dalam organisasi. Menurut W.H. Inmon, *data warehouse* memiliki empat karakteristik utama, yaitu berorientasi pada subjek (subject-oriented), terintegrasi (integrated), bersifat historis (time-variant), dan tidak mudah berubah (non-volatile) (Inmon, 1996). Artinya, data yang disimpan tidak hanya berasal dari satu sistem, tetapi dari berbagai sumber yang berbeda, digabungkan ke dalam satu struktur yang konsisten, dan digunakan untuk melihat perkembangan data dari waktu ke waktu tanpa diubah-ubah setiap hari seperti pada sistem transaksi.

Berbeda dari basis data operasional yang digunakan untuk transaksi harian seperti pembelian atau pencatatan stok, *data warehouse* lebih menekankan pada penyimpanan data dalam jumlah besar untuk dianalisis. Data yang disimpan di dalam *data warehouse* biasanya telah diringkas dan disusun dalam bentuk yang memudahkan proses analisis. Dengan bantuan teknologi seperti OLAP (Online Analytical Processing), pengguna dapat menganalisis data dari berbagai sudut pandang—misalnya berdasarkan waktu, lokasi, atau jenis layanan—untuk menemukan pola, tren, atau informasi penting lainnya.

Secara umum, *data warehouse* berfungsi sebagai fondasi bagi sistem pendukung keputusan (*decision support system*). Karena menyimpan data yang telah dikumpulkan dan digabungkan selama periode waktu yang panjang, ukuran *data warehouse* bisa sangat besar. Analisis yang dilakukan pun biasanya kompleks dan memerlukan kemampuan sistem untuk memproses permintaan data (query) secara cepat dan efisien. Oleh sebab itu, *data warehouse* menjadi bagian penting dalam strategi organisasi modern untuk memanfaatkan data sebagai dasar dalam pengambilan keputusan yang tepat dan berbasis informasi yang akurat.

2.2 OLAP vs OLTP

Sistem *Online Transaction Processing* (OLTP) dan *Online Analytical Processing* (OLAP) memiliki beberapa perbedaan mendasar dalam tujuan dan karakteristik penggunaannya.

- Orientasi Pengguna dan Sistem : Sistem OLTP (*Online Transaction Processing*) berorientasi pada pelanggan dan digunakan untuk memproses transaksi serta kueri oleh petugas, klien, atau staf teknologi informasi. Sementara itu, sistem OLAP (*Online Analytical Processing*) berorientasi pada pasar dan digunakan untuk analisis data oleh para pekerja pengetahuan seperti manajer, eksekutif, dan analis.
- Isi Data : OLTP mengelola data terkini dalam format yang sangat rinci dan mendetail. Sebaliknya, OLAP mengelola data dalam jumlah besar yang bersifat historis dan menyediakan fasilitas untuk merangkum serta mengagregasi data. Selain itu, OLAP menyimpan dan mengelola informasi pada berbagai tingkat kedalaman (granularitas), sehingga data lebih mudah digunakan dalam pengambilan keputusan yang lebih bijak dan tepat.
- Desain Database : Sistem OLTP biasanya menggunakan model data *entity-relationship* (ER) dan dirancang berdasarkan kebutuhan aplikasi operasional. Sementara sistem OLAP menggunakan model *star schema* atau *snowflake schema*, serta dirancang berdasarkan subjek atau topik tertentu, seperti penjualan, keuangan, atau produksi.
- Akses Data : Akses data pada sistem OLTP umumnya berupa transaksi pendek dan bersifat atomik (satu proses selesai tanpa tergantung proses lain). Sistem ini membutuhkan mekanisme kontrol transaksi dan pemulihan data. Sementara itu, akses pada sistem OLAP lebih banyak berupa operasi baca (*read-only*) dengan kueri yang kompleks, karena ditujukan untuk analisis dan tidak untuk memodifikasi data secara langsung.

2.3 Model Multidimensional

Model multidimensi adalah pendekatan penyimpanan dan analisis data yang memungkinkan pengguna melihat data dari berbagai perspektif, yang disebut sebagai “dimensi.” Contohnya adalah dimensi waktu, lokasi, atau jenis bahan bakar dalam konteks energi. Setiap dimensi berisi atribut yang dapat digunakan untuk pengelompokan data, seperti tahun atau bulan pada dimensi waktu, atau provinsi dan negara pada dimensi lokasi. Sementara itu, nilai-nilai

kuantitatif yang dianalisis disebut *measures*, misalnya jumlah energi yang dihasilkan, total konsumsi bahan bakar, atau biaya operasional (Atoti, 2024).

Representasi data ini biasanya divisualisasikan dalam bentuk *data cube* atau kubus data, yang memungkinkan pengguna untuk menjelajahi data melalui proses *drill-down* (melihat lebih rinci), *roll-up* (melihat lebih ringkas), *slice* (mengambil subset data berdasarkan satu dimensi), dan *dice* (mengambil subset data berdasarkan dua atau lebih dimensi). Pendekatan ini mendukung analisis yang cepat dan fleksibel, karena struktur data telah dirancang agar siap dianalisis secara agregat. Dalam dunia bisnis dan pemerintahan, model multidimensi sangat berguna untuk mendeteksi tren, membuat perbandingan, dan menghasilkan laporan analitis yang mendalam secara efisien. Dengan demikian, model ini menjadi salah satu fondasi utama dalam sistem *business intelligence* dan pengambilan keputusan berbasis data.

2.4 Hirarki

Hirarki dalam data warehouse menggambarkan struktur bertingkat dalam suatu dimensi. Misalnya, dalam dimensi waktu, pengguna dapat menelusuri data dari tahun ke kuartal, bulan, hingga hari. Hirarki ini memungkinkan pengguna melakukan *drill-down* (melihat detail lebih dalam) atau *roll-up* (melihat ringkasan data) dalam analisis OLAP (Vaisman & Zimányi, 2014).

2.5 Measures

Measures adalah elemen numerik yang dapat dianalisis dan diolah dalam data warehouse. Contoh measures dalam konteks pembangkit listrik adalah kapasitas terpasang (MW) dan estimasi produksi listrik (GWh). Measures dapat dikategorikan sebagai distributive (seperti SUM), algebraic (seperti *AVERAGE*), atau holistic (seperti *MEDIAN*). Dalam sistem seperti Atoti, measures dapat dihitung dan divisualisasikan secara langsung dalam cube analitik (Atoti, 2024).

2.6 Arsitektur Data Warehouse

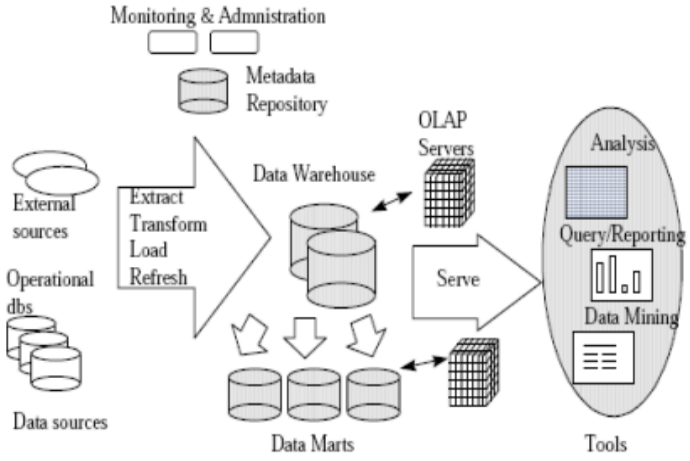


Figure 1: Data Warehousing Architecture

Gambar di atas menggambarkan arsitektur data warehouse yang terdiri dari beberapa komponen utama yang bekerja secara terintegrasi untuk mendukung proses pengumpulan, penyimpanan, dan analisis data. Proses dimulai dari sumber data eksternal maupun internal, seperti *database* operasional dan sistem eksternal lainnya. Data dari sumber-sumber ini kemudian masuk ke dalam tahap ETL (Extract, Transform, Load). Pada tahap ini, data diekstraksi dari sumbernya, dilakukan *preprocessing* dan ditransformasikan ke dalam format yang seragam, lalu dimuat ke dalam *data warehouse*. Proses refresh juga dilakukan secara periodik agar data warehouse tetap terbaru.

Setelah data tersimpan dalam data warehouse, data dapat dibagi ke dalam data marts, yaitu bagian dari data warehouse yang lebih spesifik sesuai dengan kebutuhan departemen atau divisi tertentu (misalnya keuangan, logistik, atau produksi). Data yang sudah disimpan ini kemudian disediakan untuk dianalisis oleh OLAP servers, yang menyusun data dalam bentuk multidimensi dan siap dianalisis lebih lanjut.Tahap selanjutnya adalah penggunaan tools untuk mendukung pengambilan keputusan, yang mencakup aktivitas seperti analisis data, query/reporting, dan data mining. Tool-tool ini membantu pengguna seperti manajer atau analis untuk menggali informasi penting dari data secara efisien. Selain itu, terdapat metadata repository yang menyimpan informasi mengenai struktur, sumber, dan arti dari data yang digunakan, serta komponen monitoring dan administrasi yang memastikan sistem data warehouse berjalan dengan baik dan aman.

2.7 Operasi-Operasi OLAP

OLAP memiliki sejumlah operasi dasar yang digunakan untuk manipulasi dan eksplorasi data multidimensi, yaitu:

- *Roll-up*: mengagregasi data ke level hirarki yang lebih tinggi.
- *Drill-down*: menelusuri data ke level lebih detail.
- *Slice*: mengambil subset data dari satu dimensi tertentu.
- *Dice*: mengambil sub-cube dengan dua atau lebih kondisi dimensi.
- *Pivot (rotate)*: mengubah orientasi dimensi untuk melihat data dari perspektif yang berbeda.

Operasi-operasi ini memungkinkan pengguna untuk menyesuaikan eksplorasi data sesuai kebutuhan analisis mereka.

2.8 Konsep-Konsep Komputasi Terdistribusi

Dalam arsitektur data warehouse modern, komputasi terdistribusi sangat penting karena jumlah data yang besar membutuhkan pemrosesan yang cepat dan efisien. Salah satu pendekatan yang umum digunakan adalah dengan memanfaatkan Celery dan RabbitMQ. Keduanya bekerja sama untuk menjalankan proses secara paralel dan terjadwal, sehingga sistem menjadi lebih responsif dan waktu eksekusi menjadi lebih cepat

Dalam sistem data warehouse modern, komputasi terdistribusi diperlukan untuk mengelola volume data yang besar dan proses yang kompleks secara efisien. Salah satu cara mengimplementasikan komputasi terdistribusi adalah dengan menggunakan Celery, yaitu sebuah pustaka Python yang berfungsi sebagai sistem *task queue*. Celery memungkinkan pemrosesan tugas-tugas (seperti ekstraksi data, transformasi, atau pemuatan data) dijalankan secara asinkron dan paralel di latar belakang. Ini artinya, berbagai proses bisa dilakukan secara bersamaan tanpa saling menunggu, sehingga waktu eksekusi sistem menjadi jauh lebih cepat dan tidak membebani server utama.

Agar sistem Celery dapat berjalan dengan baik, dibutuhkan komponen tambahan berupa message broker, dan salah satu yang paling banyak digunakan adalah RabbitMQ. RabbitMQ bertugas sebagai perantara komunikasi antar bagian sistem, terutama antara aplikasi utama

dan pekerja (*worker*) Celery. Ketika ada tugas yang perlu diproses, aplikasi akan mengirimkan pesan ke RabbitMQ. RabbitMQ kemudian menyimpan pesan tersebut di antrian dan mengirimkannya ke *worker* yang tersedia. Proses ini memungkinkan tugas-tugas yang masuk dapat didistribusikan dan dieksekusi secara teratur dan efisien, bahkan jika jumlah tugasnya sangat besar.

BAB III

METODE

3.1 Langkah-Langkah Instalasi Atoti, RabbitMQ, dan Celery

Proyek ini dijalankan pada sistem operasi Windows 10 menggunakan interpreter Python 3.11. Untuk membangun sistem analitik terintegrasi yang mendukung pemrosesan OLAP dan komputasi terdistribusi, digunakan pustaka Python seperti Atoti untuk eksplorasi multidimensi, RabbitMQ sebagai message broker, dan Celery sebagai task scheduler asynchronous.

1. Instalasi Atoti
    - Membuat atau Mengaktifkan Virtual Environtment

      Langkah awal dilakukan dengan membuat dan mengaktifkan lingkungan virtual (virtual environment) sebagai berikut:

```
C:\Users\Melinda Siburian>cd Downloads

C:\Users\Melinda Siburian\Downloads>python -m venv komp_dis

C:\Users\Melinda Siburian\Downloads>komp_dis\Scripts\activate

(komp_dis) C:\Users\Melinda Siburian\Downloads>
```

    - Menginstall atoti dengan perintah `pip install "atoti[jupyterlab]"`

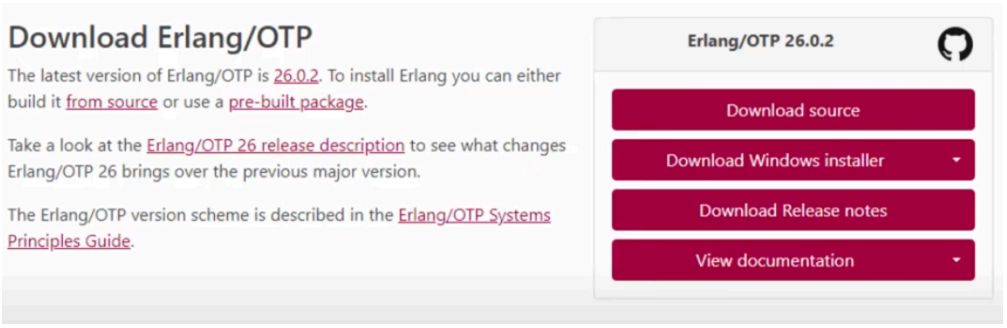
```
(komp_dis) C:\Users\Melinda Siburian\Downloads>pip install "atoti[jupyterlab]"
```

    - Cek instalasi atoti

```
(komp_dis) C:\Users\Melinda Siburian\Downloads>python
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import atoti
Welcome to Atoti 0.9.6!

By using this community edition, you agree with the license available at https://docs.activeviam.com/products/atoti/python-sdk/latest/eula.html.
Browse the official documentation at https://docs.activeviam.com/products/atoti/python-sdk.
Join the community at https://www.atoti.io/register.

Atoti collects telemetry data, which is used to help understand how to improve the product.
If you don't wish to send usage data, you can request a trial license at https://www.atoti.io/evaluation-license-request.

You can hide this message by setting the `ATOTI_HIDE_EULA_MESSAGE` environment variable to True.
>>>
```

      Instalasi atoti berhasil

2. Instalasi RabbitMQ
    - Instalasi Erlang



RabbitMQ memerlukan Erlang sebagai dependensi utama. Instalasi dilakukan melalui situs resmi:
Erlang OTP : https://www.erlang.org/downloads

Pilih versi yang kompatibel dengan RabbitMQ (biasanya versi terbaru direkomendasikan).

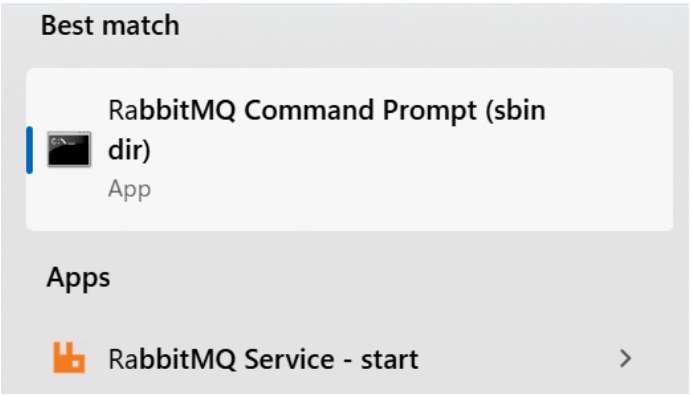
Unduh file .exe dan instal seperti biasa.

- Setelah itu install RabbitMQ

Description	Download	Signature
Installer for Windows systems (from GitHub)	rabbitmq-server-4.1.0.exe	Signature

RabbitMQ Installer for Windows
:https://www.rabbitmq.com/docs/install-windows#downloads

Unduh file .exe dan instal seperti biasa.



Jalankan RabbitMQ service dengan membuka RabbitMQ Command Prompt. Search command prompt tersebut melalui windows search. Pastikan RabbitMQ service dalam keadaan start.

```
C:\Program Files\RabbitMQ Server\rabbitmq_server-4.0.8\sbin>rabbitmqctl status
Status of node rabbit@MSI ...
[]
Runtime

OS PID: 11964
OS: Windows
Uptime (seconds): 934835
Is under maintenance?: false
RabbitMQ version: 4.0.8
RabbitMQ release series support status: see https://www.rabbitmq.com/release-information
Node name: rabbit@MSI
Erlang configuration: Erlang/OTP 27 [erts-15.2.5] [source] [64-bit] [smp:12:12] [ds:12:12
 [jit:ns]
Crypto library: OpenSSL 3.1.0 14 Mar 2023
Erlang processes: 490 used, 1048576 limit
Scheduler run queue: 1
Cluster heartbeat timeout (net_ticktime): 60
```

RabbitMQ sudah berjalan, Instalasi berhasil

Celery dan module-module python yang lain dapat diinstall dengan pip. Installasi harus di
dalam virtual environtment. Celery dan daftar module-module yang akan diinstall dituliskan
dalam sebuah file dengan nama requirements.txt.

File requirements.txt ini dapat diunduh dari tautan berikut,
https://github.com/vjanz/python-asynchronous-tasks/blob/main/requirements.txt
Installasi dilanjutkan dengan menjalankan perintah berikut pada terminal/cmd,
pip install -r requirements.txt

```
(komp_dis) C:\Users\Melinda Siburian\Downloads>pip install -r requirements.txt
```

instalasi celery selesai.

3.2 Spesifikasi Dataset

Dataset utama yang digunakan dalam proyek ini adalah Global Power Plant Database v1.3.0, yang dikembangkan oleh World Resources Institute. Dataset ini tersedia secara terbuka dengan lisensi Creative Commons [CC BY 4.0], dan dapat diakses melalui situs resmi global_power_plant_dataset.

Dataset mencakup sekitar 35.000 pembangkit listrik di 167 negara dan berisi informasi seperti:

- Nama pembangkit listrik (name)
- Kapasitas terpasang (capacity mw)

- Jenis bahan bakar utama (primary fuel)
- Tahun mulai beroperasi (commissioning year)
- Lokasi geografis (latitude, longitude)
- Estimasi produksi listrik tahunan (estimated _generation_Gwh 2013-2017)

Dataset digunakan sebagai basis untuk eksplorasi OLAP dalam konteks energi global.

3.3 Preprocessing, Processing dan Post Processing

1. Preprocessing

   Preprocessing dilakukan untuk membersihkan dan mempersiapkan data agar dapat dianalisis dengan Atoti. Beberapa tahapannya meliputi:

   - Mengecek duplikasi : Data diperiksa apakah terdapat baris duplikat berdasarkan kombinasi atribut utama. Jumlah duplikasi dihitung dan dihapus jika ditemukan. Pada dataset ini tidak ada data duplikat.
   - Mengecek dan Imputasi Missing Value

   ```
   df["other_fuel1"].fillna(df["other_fuel1"].mode()[0], inplace=True)
   df["other_fuel2"].fillna(df["other_fuel2"].mode()[0], inplace=True)
   df["other_fuel3"].fillna(df["other_fuel3"].mode()[0], inplace=True)
   ```

   Untuk kolom-kolom bertipe string seperti “other_fuel1”, “other_fuel2”, dan “other_fuel3”, dilakukan proses imputasi menggunakan nilai *mode*. Pemilihan metode ini dilakukan karena imputasi mode dianggap paling relevan dan tidak merusak distribusi kategori yang ada dalam data.

   ```
   cols = [
       "estimated_generation_gwh_2013", "estimated_generation_gwh_2014",
       "estimated_generation_gwh_2015", "estimated_generation_gwh_2016",
       "estimated_generation_gwh_2017"]
   for col in cols:
       df[col] = df[col].fillna(df[col].mean())
   ```

   Sedangkan untuk kolom-kolom numerik dilakukan proses imputasi menggunakan nilai mean. Pendekatan ini digunakan karena sederhana dan efektif dalam menjaga stabilitas nilai numerik secara umum.

- Konversi tipe data

```
# Ubah kolom menjadi datetime (hanya tahun)
df["commissioning_year"] = pd.to_datetime(df["commissioning_year"], format="%Y", errors="coerce")
df["year_of_capacity_data"] = pd.to_datetime(df["year_of_capacity_data"], format="%Y", errors="coerce")
```

```
df["capacity_gw"] = df["capacity_mw"] / 1000
df["capacity_tw"] = df["capacity_mw"] / 1_000_000
```

```
df["capacity_gw"] = df["capacity_gw"].astype(float)
df["capacity_tw"] = df["capacity_tw"].astype(float)
df["capacity_mw"] = df["capacity_mw"].astype(float)
```

Untuk analisis lebih lanjut, kolom capacity_mw diubah kesatuan gigawatt (gw) dan terawatt (tw) untuk mempermudah analisis. Setelah itu, tipe kolom dikonversi ke tipe float.

2. Processing

```
[14]: session = tt.Session.start()

[15]: df1=df[["country_long", "name", "gppd_idnr","capacity_mw", "primary_fuel", "country",
              "estimated_generation_gwh_2013", "estimated_generation_gwh_2014", "estimated_generation_gwh_2015",
             "estimated_generation_gwh_2016","estimated_generation_gwh_2017","other_fuel1","other_fuel2",
              "other_fuel3","latitude","longitude","capacity_gw","capacity_tw"]]

[16]: # 1. Dimensi Pembangkit Listrik
dim_name = df1[["gppd_idnr", "name","latitude","longitude"]].drop_duplicates()
dim_name_table = session.read_pandas(dim_name, table_name="Dim_PowerPlant", keys=["gppd_idnr"])

[17]: dim_country = df1[["country_long","country"]].drop_duplicates().reset_index(drop=True)
dim_country["id_country"] = dim_country.index + 1
dim_country_table = session.read_pandas(dim_country, table_name="Dim_Country", keys=["id_country"])

[18]: dim_fuel = df1[["primary_fuel","other_fuel1","other_fuel2","other_fuel3"]].drop_duplicates().reset_index(drop=True)
dim_fuel["id_fuel"] = dim_fuel.index + 1
dim_fuel_table = session.read_pandas(dim_fuel, table_name="Dim_Fuel", keys=["id_fuel"])
```

Proses dimulai dengan inisialisasi sesi Atoti (session = tt.Session.start()), lalu data utama (df1) disiapkan dengan memilih kolom-kolom penting terkait pembangkit listrik, termasuk kapasitas, lokasi, dan estimasi produksi listrik tahunan yang dibutuhkan untuk analisis selanjutnya.

Selanjutnya, dibentuk tabel dimensi seperti Dim_PowerPlant (berisi informasi nama dan lokasi pembangkit), Dim_Country (berisi informasi negara), dan Dim_Fuel (berisi jenis bahan bakar utama dan tambahan). Setiap dimensi dibuat unik menggunakan drop_duplicates() dan diberi ID untuk keperluan relasi.

[19]:
```
# Gabung ID Country
fact_df = df1.merge(dim_country, on="country_long", how="left")

# Gabung ID Fuel
fact_df = fact_df.merge(dim_fuel, on="primary_fuel", how="left")

# Buat tabel fakta dengan hanya ID dan metrik numerik
fact_generation = fact_df[["id_country", "id_fuel", "gppd_idnr", "capacity_mw", "country_long", "primary_fuel",
                           "name","estimated_generation_gwh_2013", "estimated_generation_gwh_2014",
                           "estimated_generation_gwh_2015", "estimated_generation_gwh_2016",
                           "estimated_generation_gwh_2017","capacity_gw","capacity_tw"]]
```

[20]:
```
fact_table = session.read_pandas(fact_generation,
                                 table_name='Fact_Power_Generation',
                                 data_types={
                                     'id_country': tt.type.INT,
                                     'id_fuel': tt.type.INT,
                                     'gppd_idnr': tt.type.STRING,
                                     'country_long': tt.type.STRING,
                                     'primary_fuel': tt.type.STRING,
                                     'name': tt.type.STRING,
                                     'capacity_mw': tt.type.FLOAT,
```

```
fact_table.join(dim_name_table)
fact_table.join(dim_country_table)
fact_table.join(dim_fuel_table)
```

```
session.tables.schema
```

Fact_Power_Generation			
_	long	PK	id_country
_	long	PK	id_fuel

Kemudian, dilakukan penggabungan (join) antara data utama (fact_df) dengan tabel dimensi untuk menambahkan id_country dan id_fuel. Dari hasil tersebut, dibentuklah tabel fakta Fact_Power_Generation, yang hanya berisi metrik numerik dan ID dari dimensi sebagai kunci utama. Tabel ini dimuat ke Atoti dengan spesifikasi tipe data untuk setiap kolom.

```
cube = session.create_cube(fact_table)
h = cube.hierarchies
l = cube.levels
m = cube.measures
```

Langkah terakhir adalah membangun cube OLAP dari tabel fakta menggunakan session.create_cube(). Proses ini menghasilkan objek cube yang dapat digunakan untuk eksplorasi data multidimensi melalui hierarchies, levels, dan measures.

3. Postprocessing

Melakukan kueri cube

Pertanyaan 1

"Berapa total estimasi pembangkitan listrik (dalam GWh) berdasarkan jenis bahan bakar (fuel) pada tahun 2017 di setiap negara?"

```
m["Total_Generation_2017"] = tt.agg.sum(fact_table["estimated_generation_gwh_2017"])
```

```
print("Hierarchies:", list(h.keys()))
print("Levels:", list(l.keys()))
print("Measures:", list(m.keys()))
```

Postproses data warehouse pada kasus ini melibatkan beberapa langkah penting untuk menjawab pertanyaan analitis berdasarkan data kubus OLAP. Pertama, dilakukan agregasi data dari tabel fakta, yaitu total estimasi pembangkitan listrik tahun 2017 (estimated_generation_gwh_2017), menggunakan fungsi agg.sum() untuk menghasilkan sebuah *measure* baru bernama Total_Generation_2017.

```
result = cube.query(
    m["Total_Generation_2017"],
    levels=[
        l[("Dim_Country", "country_long", "country_long")],
        l[("Dim_Fuel", "primary_fuel", "primary_fuel")]
    ]
)

result
```

Selanjutnya, kueri terhadap cube dilakukan dengan mengambil data berdasarkan *level* dari dua dimensi: negara (Dim_Country) dan jenis bahan bakar (Dim_Fuel). Hasil kueri ini memberikan agregasi total pembangkitan listrik per jenis bahan bakar di tiap negara.

```
[29]: session.widget
```

```
# Labeling
plt.xticks(x + width * (len(fuels) - 1) / 2, countries, rotation=45, ha='right')
plt.ylabel("Total Generation (GWh)")
plt.xlabel("Country")
plt.title("Top 10 Countries by Total Estimated Generation (2017) by Fuel Type")
plt.legend(title="Primary Fuel", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

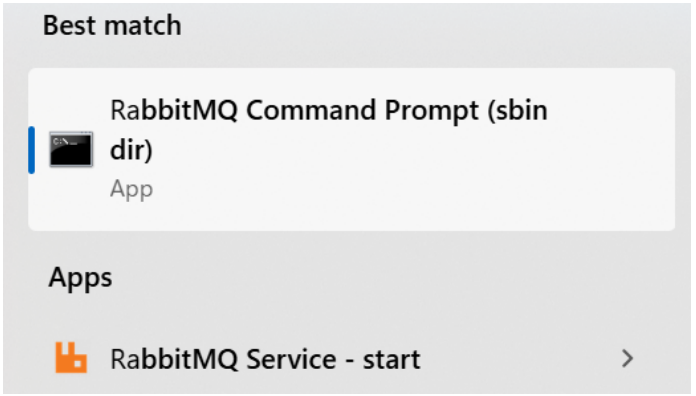
Terakhir, data divisualisasikan menggunakan matplotlib dalam bentuk grafik batang yang menampilkan 10 negara teratas berdasarkan total pembangkitan listrik, dengan pewarnaan berbeda untuk setiap jenis bahan bakar. Proses ini menunjukkan alur dari pemodelan hingga analisis visual data berbasis OLAP.

3.4 Komputasi Terdistribusi

Untuk menyimpan data dan plot yang dihasilkan dari cube.query() di JupyterLab, kami menerapkan komputasi terdistribusi menggunakan RabbitMQ dan Celery guna meningkatkan efisiensi pemrosesan dan pengelolaan data berskala besar. Komputasi terdistribusi memungkinkan pemrosesan dilakukan secara paralel di beberapa node atau core (dalam hal ini melalui terminal\command prompt), sehingga mempercepat eksekusi query agregasi yang kompleks. Hasil dari query kemudian dapat disimpan dalam format seperti CSV sementara visualisasi seperti grafik dan plot disimpan dalam format gambar (PNG).

Berikut langkah-langkah yang dilakukan dalam komputasi terdistribusi dengan RabbitMQ dan Celery :

1. Status RabbitMQ Service dalam keadaan start



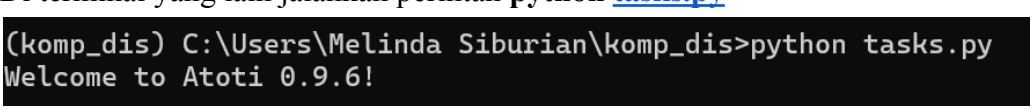
Jalankan RabbitMQ service dengan membuka RabbitMQ Command Prompt. Search command prompt tersebut melalui windows search. Pastikan RabbitMQ service dalam keadaan start.

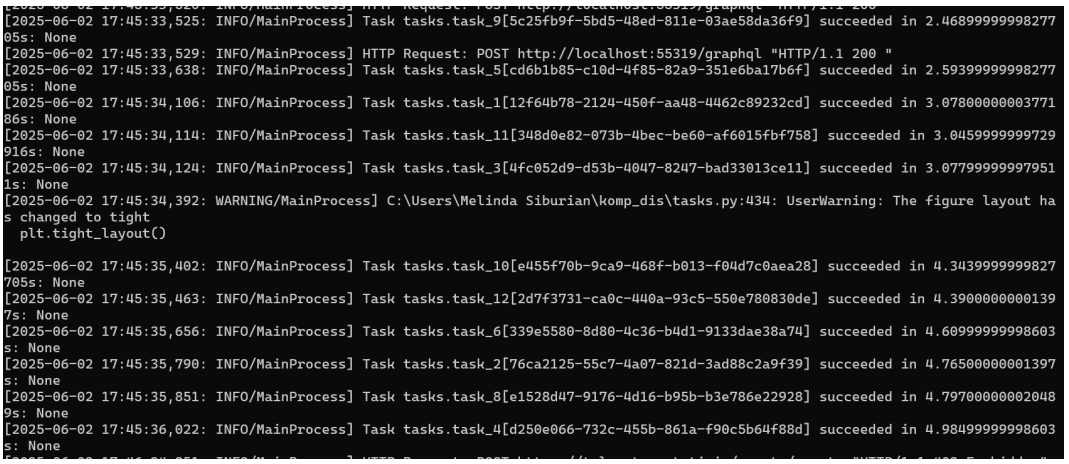
```
C:\Program Files\RabbitMQ Server\rabbitmq_server-4.0.8\sbin>rabbitmqctl status
Status of node rabbit@MSI ...
[]
Runtime

OS PID: 11964
OS: Windows
Uptime (seconds): 934835
Is under maintenance?: false
RabbitMQ version: 4.0.8
RabbitMQ release series support status: see https://www.rabbitmq.com/release-information
Node name: rabbit@MSI
Erlang configuration: Erlang/OTP 27 [erts-15.2.5] [source] [64-bit] [smp:12:12] [ds:12:12
 [jit:ns]
Crypto library: OpenSSL 3.1.0 14 Mar 2023
Erlang processes: 490 used, 1048576 limit
Scheduler run queue: 1
Cluster heartbeat timeout (net_ticktime): 60
```

RabbitMQ sebagai Message Broker yang berfungsi mengirimkan task dari terminal satu ke terminal yang lain sudah siap untuk digunakan.

2. Siapkan dua terminal dengan virtual environment dan directory yang sama dimana atoti dan rabbitmq berada
3. Langkah selanjutnya adalah menjalankan worker celery dengan perintah berikut **celery -A tasks worker -l info --pool=threads.** Perintah ini dijalankan di salah satu terminal.

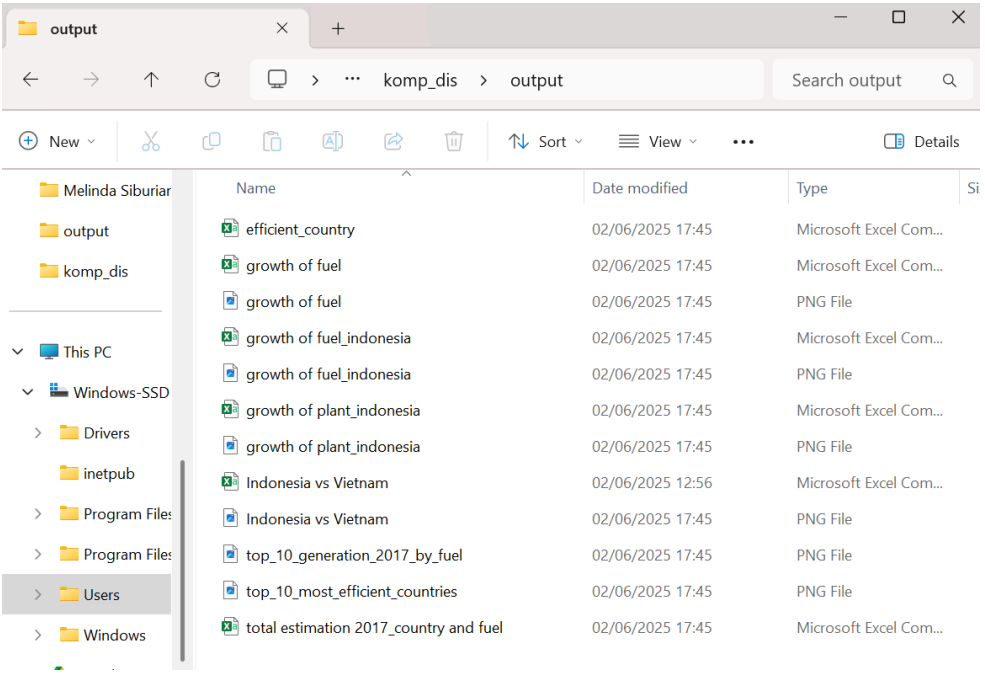
Di terminal yang lain jalankan perintah **python** tasks.py





Dengan kombinasi RabbitMQ dan Celery tasks dapat diselesaikan dengan cepat dan efisien

4. Hasil dari 12 tasks ini disimpan dalam folder output yang sudah disiapkan sebelumnya.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Preprocessing

Setelah dilakukan imputasi dengan nilai modus dan median didapatkan tidak ada lagi missing value pada kolomm-kolom yang dibutuhkan untuk analisis

```
# Cek apakah masih ada NaN
print(df.isnull().sum())
```

```
country                          0
country_long                     0
name                             0
gppd_idnr                        0
capacity_mw                      0
latitude                         0
longitude                        0
primary_fuel                     0
other_fuel1                      0
other_fuel2                      0
other_fuel3                      0
estimated_generation_gwh_2013    0
estimated_generation_gwh_2014    0
estimated_generation_gwh_2015    0
estimated_generation_gwh_2016    0
estimated_generation_gwh_2017    0
estimated_generation_note_2013   0
estimated_generation_note_2014   0
estimated_generation_note_2015   0
estimated_generation_note_2016   0
estimated_generation_note_2017   0
capacity_gw                      0
capacity_tw                      0
```

4.2 Hasil Cube (Hierarki, Measure, Level)

1. Hierarki

```
[24]: h
```

```
[24]: Dimensions
  Dim_Country
    country [] 1 item
      0 "country"
    country_long [] 1 item
      0 "country_long"
  Dim_Fuel
    other_fuel1 [] 1 item
      0 "other_fuel1"
    other_fuel2 [] 1 item
      0 "other_fuel2"
    other_fuel3 [] 1 item
      0 "other_fuel3"
    primary_fuel [] 1 item
      0 "primary_fuel"
  Dim_PowerPlant
    name [] 1 item
      0 "name"
Fact_Power_Generation
  country_long [] 1 item
    0 "country_long"
  gppd_idnr [] 1 item
    0 "gppd_idnr"
  id_country [] 1 item
    0 "id_country"
  id_fuel [] 1 item
    0 "id_fuel"
  name [] 1 item
    0 "name"
  primary_fuel [] 1 item
    0 "primary_fuel"
```

2. Measure

```
[25]: m
```

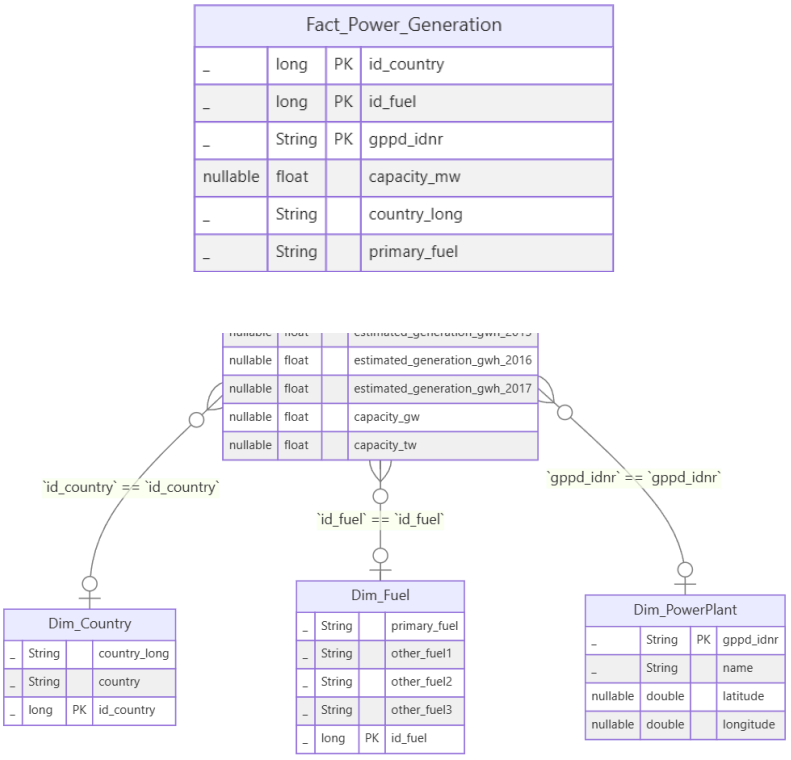
```
[25]: Measures
  capacity_gw.MEAN
  capacity_gw.SUM
  capacity_mw.MEAN
  capacity_mw.SUM
  capacity_tw.MEAN
  capacity_tw.SUM
  contributors.COUNT
  estimated_generation_gwh_2013.MEAN
  estimated_generation_gwh_2013.SUM
  estimated_generation_gwh_2014.MEAN
  estimated_generation_gwh_2014.SUM
  estimated_generation_gwh_2015.MEAN
  estimated_generation_gwh_2015.SUM
  estimated_generation_gwh_2016.MEAN
  estimated_generation_gwh_2016.SUM
  estimated_generation_gwh_2017.MEAN
  estimated_generation_gwh_2017.SUM
```

3. Level

```
[79]: Levels
  country (Dim_Country/country)
    dimension "Dim_Country"
    hierarchy "country"
    data type "String"
    order "NaturalOrder"
  country_long (Dim_Country/country_long)
    dimension "Dim_Country"
    hierarchy "country_long"
    data type "String"
    order "NaturalOrder"
  country_long (Fact_Power_Generation/country_long)
    dimension "Fact_Power_Generation"
    hierarchy "country_long"
    data type "String"
    order "NaturalOrder"
  gppd_idnr (Fact_Power_Generation/gppd_idnr)
    dimension "Fact_Power_Generation"
    hierarchy "gppd_idnr"
    data type "String"
    order "NaturalOrder"
  id_country (Fact_Power_Generation/id_country)
    dimension "Fact_Power_Generation"
    hierarchy "id_country"
    data type "long"
    order "NaturalOrder"
  id_fuel (Fact_Power_Generation/id_fuel)
    dimension "Fact_Power_Generation"
    hierarchy "id_fuel"
    data type "long"
    order "NaturalOrder"
  name (Dim_PowerPlant/name)
    dimension "Dim_PowerPlant"
    hierarchy "name"
    data type "String"
    order "NaturalOrder"
  name (Fact_Power_Generation/name)
    dimension "Fact_Power_Generation"
    hierarchy "name"
    data type "String"
    order "NaturalOrder"
```

- **other_fuel1 (Dim_Fuel/other_fuel1)**
  - **dimension** "Dim_Fuel"
  - **hierarchy** "other_fuel1"
  - **data type** "String"
  - **order** "NaturalOrder"
- **other_fuel2 (Dim_Fuel/other_fuel2)**
  - **dimension** "Dim_Fuel"
  - **hierarchy** "other_fuel2"
  - **data type** "String"
  - **order** "NaturalOrder"
- **other_fuel3 (Dim_Fuel/other_fuel3)**
  - **dimension** "Dim_Fuel"
  - **hierarchy** "other_fuel3"
  - **data type** "String"
  - **order** "NaturalOrder"
- **primary_fuel (Dim_Fuel/primary_fuel)**
  - **dimension** "Dim_Fuel"
  - **hierarchy** "primary_fuel"
  - **data type** "String"
  - **order** "NaturalOrder"
- **primary_fuel (Fact_Power_Generation/primary_fuel)**
  - **dimension** "Fact_Power_Generation"
  - **hierarchy** "primary_fuel"

4. Skema Join



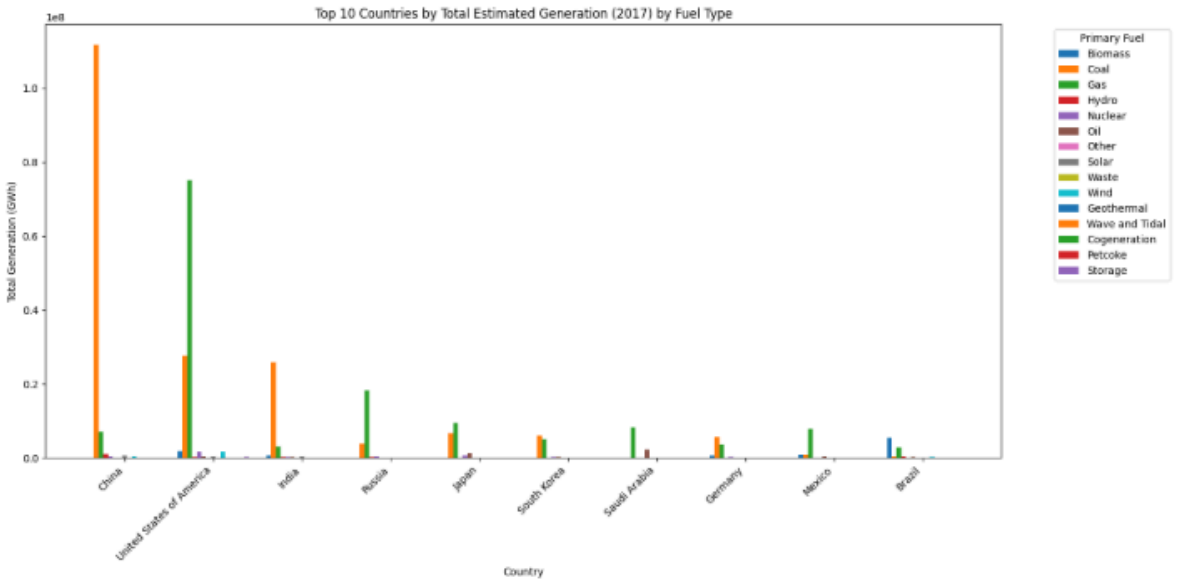
4.3 Analisis pertanyaan 1

“Berapa total estimasi pembangkit listrik (dalam GWh) berdasarkan jenis bahan bakar (fuel) pada tahun 2017 di setiap negara?”

Untuk menjawab pertanyaan ini, dilakukan query terhadap cube Atoti dengan level dimensi negara dan jenis bahan bakar, serta measure "Total_Generation_2017". Berikut adalah potongan kode yang digunakan:

```
result = cube.query(
    m["Total_Generation_2017"],
    levels=[
        l[("Dim_Country", "country_long", "country_long")],
        l[("Dim_Fuel", "primary_fuel", "primary_fuel")]])
```

Measure Total_Generation_2017 menghitung jumlah pembangkit listrik (GWh) yang diperirakan pada tahun 2017, diolah berdasarkan dimensi country_long (negara) dan primary_fuel (jenis bahan bakar).



Dari visualisasi diatas menunjukkan bahwa negara dengan pembangkit terbesar antara lain adalah China menduduki peringkat sebagai negara dengan estimasi total pembangkit listrik terbesar. Kontribusi dominan berasal dari batubara (coal), yang menjadi sumber utama pembangkit listrik di negara tersebut. Ini sejalan dengan profil industri energi China yang masih sangat bergantung pada bahan bakar fosil, meskipun investasi pada energi terbarukan mulai tumbuh. Di peringkat berikutnya, USA menunjukkan profil energi yang lebih beragam, dengan kontribusi signifikan dari gas alam, diikuti oleh tenaga nuklir (nuclear) dan angin

(wind). Hal ini mencerminkan kebijakan transisi energi yang lebih progresif dibandingkan beberapa negara lainnya.

India juga masuk dalam daftar dengan produksi energi tinggi, yang sebagian besar masih ditopang oleh batu bara (coal). Meski demikian, kontribusi dari energi surya (solar) dan air (hydro) mulai terlihat, menandakan arah kebijakan yang perlahan menuju diversifikasi energi. Sementara itu, negara seperti Brazil menunjukkan ketergantungan besar pada pembangkit listrik berbasis air (hydro), yang wajar mengingat potensi sungai dan bendungan yang melimpah di negara tersebut. Di sisi lain, Jerman menampilkan bauran energi yang menarik, dengan peran besar dari angin dan surya, mencerminkan implementasi kebijakan Energiewende yang mendorong pergeseran menuju energi bersih.

Negara seperti Rusia, Kanada, dan Jepang juga muncul dalam daftar, masing-masing dengan profil energi yang khas. Rusia mengandalkan gas alam dan hidro, sedangkan Kanada memaksimalkan potensi hydro. Jepang sendiri menunjukkan dominasi pembangkitan dari gas dan batu bara, namun dengan kontribusi energi terbarukan yang mulai meningkat. Meskipun Indonesia tidak termasuk dalam sepuluh besar negara dengan estimasi pembangkitan tertinggi secara total, kontribusinya masih signifikan terutama dalam kategori batu bara, geothermal, dan hydro. Hal ini mencerminkan karakteristik energi nasional yang bertumpu pada kekayaan sumber daya alam, namun menghadapi tantangan besar dalam melakukan transisi ke sumber energi bersih.

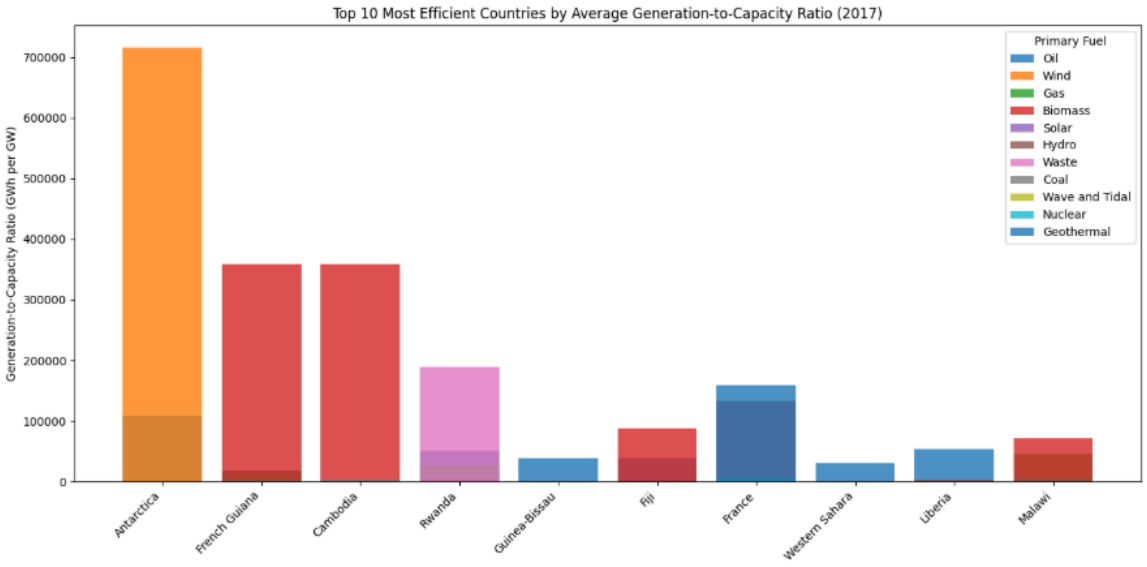
Secara keseluruhan, visualisasi ini menunjukkan perbedaan pendekatan energi di setiap negara. Negara maju cenderung telah memulai transisi ke energi bersih dengan campuran sumber daya yang lebih seimbang, sementara negara berkembang masih bergantung pada bahan bakar fosil karena ketersediaan dan efisiensi ekonominya.

4.4 Analisis pertanyaan 2

“Negara mana yang paling efisien menghasilkan listrik dilihat dari perbandingan estimated generation dan kapasitas, dan menggunakan bahan bakar apa?”

```
result = cube.query(
    m["Total_Generation_2017"], m["Total_Capacity_GW"], m["Generation_to_Capacity_Ratio"],
    levels= [
        l[("Dim_Country", "country_long", "country_long")],
        l[("Dim_Fuel", "primary_fuel", "primary_fuel")]])
```

Measure ini merepresentasikan seberapa optimal suatu negara memanfaatkan kapasitas pembangkitnya untuk menghasilkan listrik. Nilai yang tinggi menunjukkan bahwa kapasitas yang tersedia digunakan secara efisien untuk pembangkitan energi.

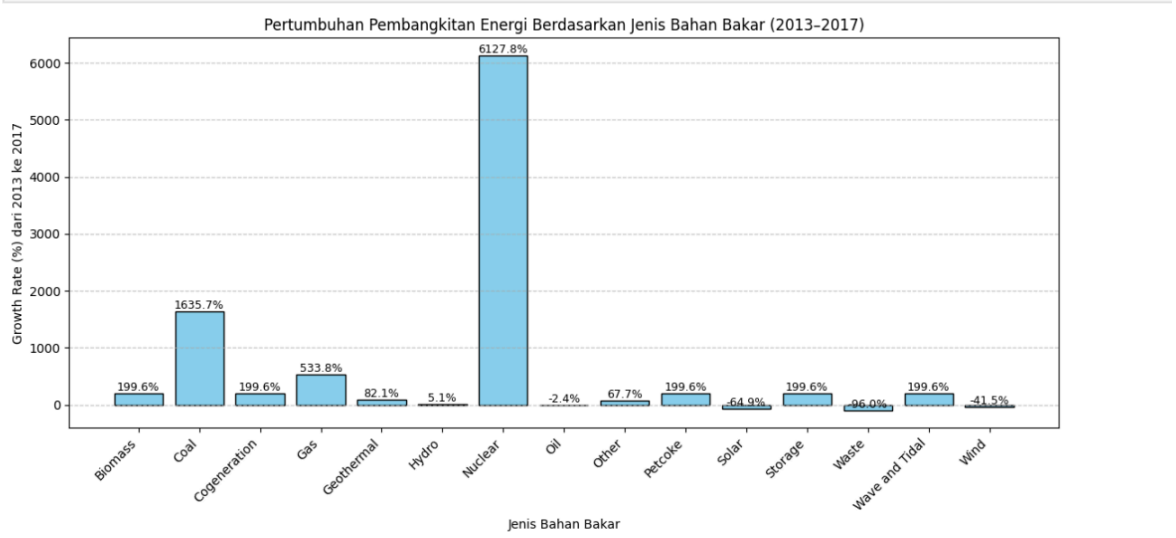


Dari Grafik, terlihat bahwa Antarctica menempati posisi teratas dengan rasio efisiensi tinggi, didominasi oleh pembangkit tenaga angin. Posisi ini cukup unik karena walaupun kapasitas total di wilayah ini relatif kecil, pemanfaatannya sangat maksimal, menghasilkan output listrik yang sangat tinggi dibandingkan kapasitas yang tersedia. Disusul oleh French Guiana dan Cambodia, keduanya menunjukkan rasio efisiensi tinggi dengan dominasi pembangkit biomassa. Hal ini menunjukkan bahwa di negara-negara tersebut, sumber energi terbarukan seperti biomassa dimanfaatkan secara optimal dalam sistem pembangkit listrik mereka. Negara-negara lain dalam daftar seperti Rwanda dengan efisiensi tinggi dari pembangkit berbahan waste dan solar, serta France yang menunjukkan kontribusi besar dari pembangkit nuklir dan geothermal, menambah keragaman sumber daya yang digunakan secara efisien. Visualisasi ini menekankan bahwa efisiensi pembangkitan energi tidak hanya ditentukan oleh skala produksi atau kapasitas total, tetapi sangat dipengaruhi oleh jenis bahan bakar utama yang digunakan serta pengelolaan sistem pembangkitan di masing-masing negara. Negara dengan teknologi pembangkit yang tepat dan sistem distribusi yang baik mampu menghasilkan energi dalam jumlah besar meskipun dengan kapasitas terbatas, sehingga menghasilkan rasio efisiensi yang tinggi.

4.5 Analisis pertanyaan 3

“Jenis bahan bakar mana yang menunjukkan pertumbuhan tertinggi dalam pembangkitan selama 5 tahun?”

```
result_growth_by_fuel = cube.query(
    m["Total_Generation_2013"],
    m["Total_Generation_2017"],
    m["Generation_Growth_Rate"],
    levels=[l[("Dim_Fuel", "primary_fuel", "primary_fuel")]]).reset_index()
```



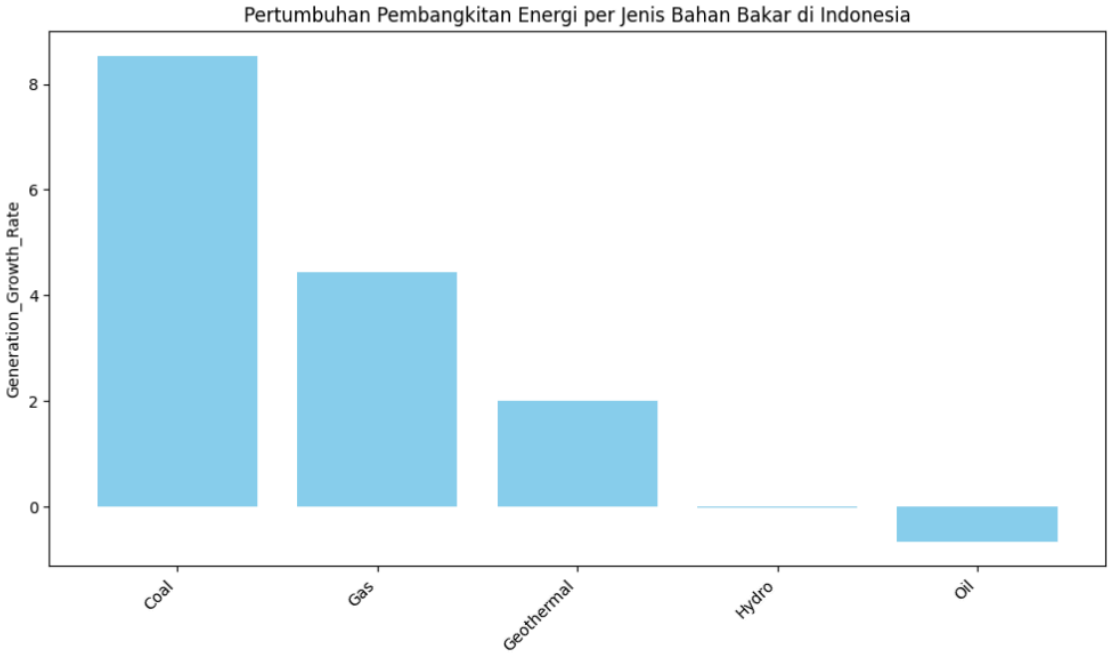
Dari grafik didapatkan bahwa, Hasil yang paling mencolok adalah pada bahan bakar nuklir, yang mencatatkan pertumbuhan luar biasa sebesar 6127,8%, menjadikannya sumber energi dengan peningkatan paling drastis selama periode tersebut. Ini mungkin mencerminkan kebijakan baru di beberapa negara untuk beralih ke energi rendah karbon. Selanjutnya, batubara (coal) juga menunjukkan lonjakan signifikan dengan 1635,7%, disusul oleh gas dengan pertumbuhan sebesar 533,8%. Ketiganya merupakan bahan bakar fosil dan termal yang sering digunakan untuk kebutuhan energi besar, terutama di negara-negara berkembang yang sedang mengalami ekspansi infrastruktur energi.

Sebaliknya, beberapa jenis energi terbarukan seperti solar dan wind justru mengalami penurunan, masing-masing sebesar -64,9% dan -41,5%, yang cukup mengherankan mengingat tren global menuju energi bersih. Hal ini bisa disebabkan oleh ketidakstabilan proyek atau kurangnya investasi pada teknologi dan jaringan distribusi yang mendukung energi surya dan angin selama rentang waktu tersebut. Penurunan lain juga terjadi pada waste (-96,0%) dan oil (-2,4%), yang dapat dikaitkan dengan pergeseran kebijakan energi dan tantangan logistik atau lingkungan.

4.6 Analisis pertanyaan 4

“Jenis bahan bakar apa yang menunjukkan pertumbuhan tertinggi di Indonesia?”

```
result_growth_indonesia = cube.query(
    m["Total_Generation_2013"],
    m["Total_Generation_2017"],
    m["Generation_Growth_Rate"],
    levels=[l[("Dim_Fuel", "primary_fuel", "primary_fuel")]],
    filter=(l[("Dim_Country", "country_long", "country_long")] == "Indonesia")
).sort_values("Generation_Growth_Rate", ascending=False)
```



Berdasarkan grafik di atas, terlihat bahwa batubara (coal) merupakan jenis bahan bakar yang menunjukkan pertumbuhan tertinggi dalam pembangkitan energi di Indonesia selama periode 2013 hingga 2017. Nilai *growth rate*-nya jauh lebih besar dibandingkan bahan bakar lainnya, yang menunjukkan bahwa pembangkit listrik berbahan bakar batubara mengalami peningkatan produksi listrik yang sangat pesat. Hal ini kemungkinan besar karena pembangunan pembangkit baru atau peningkatan kapasitas pembangkit batubara yang sudah ada, seiring meningkatnya kebutuhan listrik nasional.

Di posisi kedua, gas juga mengalami pertumbuhan yang cukup tinggi, meskipun tidak sebesar batubara. Ini menunjukkan bahwa selain batubara, gas masih menjadi alternatif utama untuk pembangkitan listrik karena ketersediaannya yang cukup stabil dan penggunaannya yang lebih ramah lingkungan dibanding batubara.

Sementara itu, geothermal (panas bumi) juga menunjukkan pertumbuhan, walaupun tidak terlalu besar. Hal ini cukup positif karena geothermal merupakan sumber energi terbarukan yang potensinya sangat besar di Indonesia. Namun, pertumbuhannya yang masih rendah bisa jadi disebabkan oleh tantangan dalam pengembangan teknologinya yang kompleks dan mahal.

Untuk hydro (air), pertumbuhannya sangat kecil, hampir mendekati nol, dan oil (minyak) justru menunjukkan pertumbuhan negatif. Artinya, penggunaan minyak sebagai sumber energi pembangkit cenderung berkurang. Ini bisa jadi karena biaya operasionalnya yang tinggi dan lebih tidak efisien dibandingkan sumber lain seperti batubara dan gas.

Secara keseluruhan, grafik ini memperlihatkan bahwa Indonesia masih sangat bergantung pada energi fosil, terutama batubara dan gas, untuk memenuhi kebutuhan listriknya. Sementara itu, penggunaan energi bersih atau terbarukan seperti hydro dan geothermal masih belum berkembang secara maksimal dalam periode tersebut.

4.7 Analisis pertanyaan 5

“Nama pembangkit listrik di Indonesia dengan pertumbuhan tercepat selama 5 tahun di Indonesia?”

```
result_growth_by_plant = cube.query(
    m["Total_Generation_2013"],
    m["Total_Generation_2017"],
    m["Generation_Growth_Rate"],
    levels=[
        l[("Dim_PowerPlant", "name", "name")]
    ],
    filter=(l[("Dim_Country", "country_long", "country_long")] == "Indonesia")).sort_values("Generation_Growth_Rate", ascending=False)
```

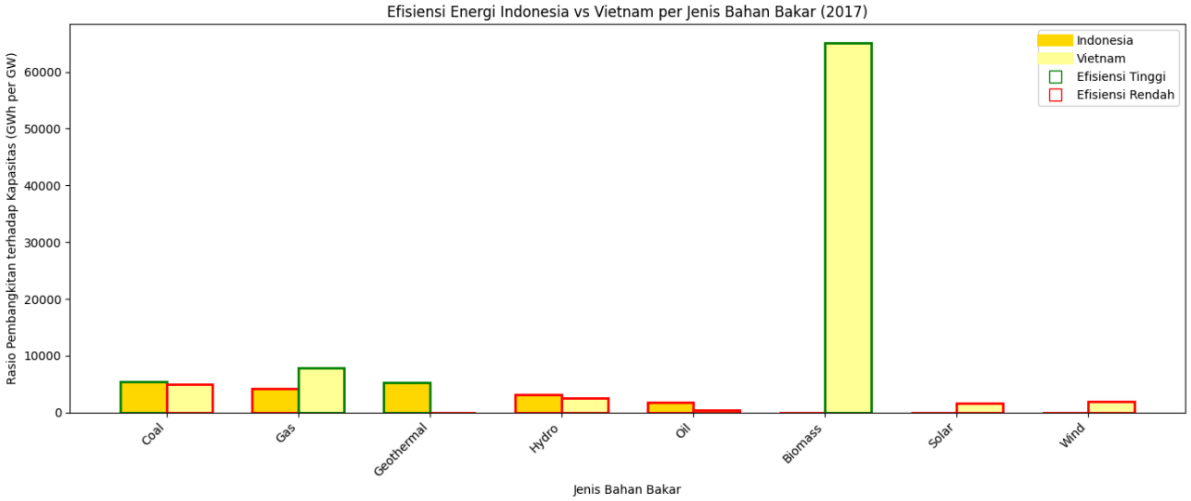
Dari grafik yang ditampilkan, terlihat bahwa PLTU Paiton I Unit 7 & 8 menjadi pembangkit listrik dengan pertumbuhan paling cepat di Indonesia selama lima tahun terakhir (2013–2017). Kenaikannya jauh lebih tinggi dibandingkan pembangkit lainnya, menunjukkan bahwa mungkin ada perluasan unit atau peningkatan kapasitas besar-besaran di sana. Di urutan kedua, ada PLTU Suralaya, yang juga mengalami peningkatan cukup besar dalam jumlah listrik yang dihasilkan.

Beberapa pembangkit lain seperti PLTU Tanjung Jati B, baik unit utama maupun unit ekspansinya, serta Cilacap Sumber Power Station dan PLTGU Gresik, juga termasuk dalam daftar 10 besar pembangkit dengan pertumbuhan tertinggi. Kebanyakan dari pembangkit ini menggunakan bahan bakar batubara atau gas, yang memang masih jadi andalan di Indonesia karena bisa menghasilkan energi dalam jumlah besar dan cocok untuk kebutuhan listrik yang terus meningkat.

Hal menarik dari grafik ini adalah semua pembangkit yang masuk daftar merupakan jenis PLTU atau PLTGU, yang artinya masih berbasis energi fosil dan termal. Tidak ada pembangkit dari energi terbarukan seperti tenaga surya atau angin yang masuk 10 besar. Ini bisa menunjukkan kalau pada periode itu, Indonesia masih fokus pada sumber energi konvensional untuk memenuhi kebutuhan listriknya, sementara pengembangan energi bersih mungkin belum maksimal.

4.8 Analisis Pertanyaan 6

“Bagaimana estimasi perbandingan efisiensi pembangkitan listrik (generation-to-capacity ratio) antara Indonesia dan Vietnam berdasarkan jenis bahan bakar pada tahun 2017?”



Grafik ini membandingkan efisiensi energi Indonesia dan Vietnam berdasarkan jenis bahan bakar pada tahun 2017. Sumbu vertikal menunjukkan rasio peningkatan terhadap kapasitas (kWh per GW), yang merupakan indikator efisiensi energi, sedangkan sumbu horizontal menunjukkan jenis bahan bakar seperti batubara (Coal), gas, panas bumi (Geothermal), hydro, minyak (Oil), biomassa, surya (Solar), dan angin (Wind).

Warna oranye mewakili data dari Indonesia dan kuning untuk Vietnam. Garis tepi hijau mengindikasikan efisiensi tinggi, sedangkan garis tepi merah menunjukkan efisiensi rendah.

Dari grafik ini terlihat bahwa Vietnam menunjukkan efisiensi yang jauh lebih tinggi pada biomassa dibandingkan Indonesia, dengan rasio yang sangat mencolok. Untuk bahan bakar lain seperti gas dan geothermal, Vietnam juga menunjukkan efisiensi yang relatif lebih tinggi dibandingkan Indonesia. Di sisi lain, Indonesia dan Vietnam memiliki efisiensi rendah pada sumber energi terbarukan seperti angin dan surya, yang ditandai dengan garis tepi merah. Secara keseluruhan, grafik ini menggambarkan bahwa Vietnam lebih unggul dalam efisiensi energi untuk sebagian besar jenis bahan bakar dibandingkan Indonesia pada tahun 2017.

V. REFERENSI

Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems*. Data-Centric Systems and Applications. Springer.

Inmon, W.H. (1996) Building the Data Warehouse, Second Edition, New York: John Wiley & Sons

Byers, L. et al. (2021). *A Global Database of Power Plants*. World Resources Institute. [Online]. Available: https://datasets.wri.org/dataset/globalpowerplantdatabase

Yin, L., Byers, L., Malaguzzi Valeri, L., & Friedrich, J. (2020). *Estimating Power Plant Generation in the Global Power Plant Database*. WRI Technical Note.

Atoti. (2024). *Multidimensional Concepts*. Retrieved from https://docs.activeviam.com/products/atoti/python-sdk/latest

Celery Project. (2025). *Celery Documentation*. https://docs.celeryq.dev
RabbitMQ. (2025). *RabbitMQ: Messaging that just works*. https://www.rabbitmq.com

Python Software Foundation. (2024). *pandas: Data analysis and manipulation tool*. https://pandas.pydata.org