

**Rancangan dan Implementasi Arsitektur Data Warehouse pada Industri
Lingkungan & Sustainability pada perusahaan ElSusEnv**

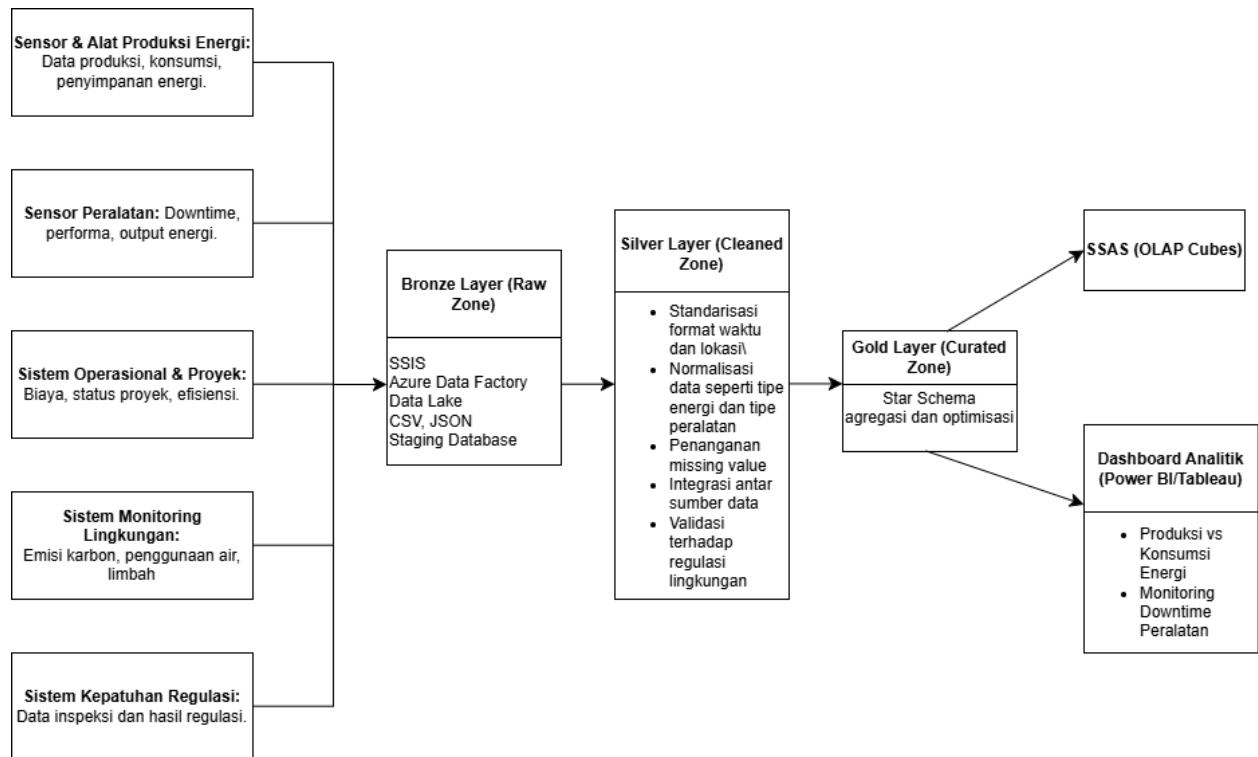


Disusun Oleh:

Muhammad Farhan	121450044
Mayada	121450145
Khoirul Anam	122450039
Alvia Asrinda Br Ginting	122450077
Syalaisha Andina Putrinsyah	122450121

**PROGRAM STUDI SAINS DATA
FAKULTAS SAINS
INSTITUT TEKNOLOGI SUMATERA
2025**

Bagian 1 Alur Aliran Data



Pada diagram alur aliran data menunjukkan bagaimana data yang dikumpulkan dari berbagai sumber diproses dan dibuat menjadi informasi yang dapat digunakan oleh stakeholder perusahaan energi terbarukan. Proses dimulai dengan pengumpulan data mentah dari beberapa sistem utama. Ini termasuk sensor dan alat produksi energi (data tentang produksi, konsumsi, dan penyimpanan energi), sensor peralatan (data tentang downtime, kinerja, dan output energi), sistem operasional dan proyek (data tentang biaya dan efisiensi proyek), dan sistem kepatuhan regulasi (data tentang hasil inspeksi dan kepatuhan regulasi).

Pertama, data disimpan dalam Lapisan Bronze (Raw Zone) menggunakan platform seperti SSIS, Data Factory, Azure, atau Data Lake dalam format CSV atau JSON, atau melalui staging database. Kemudian, Lapisan Silver (Cleaned Zone) menangani standarisasi format waktu dan lokasi, normalisasi tipe data, integrasi antar sumber data, dan validasi peraturan lingkungan yang berlaku.

Data yang telah disiapkan dan dibersihkan kemudian dikirim ke Lapisan Emas (Zona Kuadrat). Pada tahap ini, data disusun dalam skema bintang dan dilakukan agregasi dan optimisasi untuk membuatnya mudah diakses untuk analisis lebih lanjut. Output Gold Layer didistribusikan ke dua jalur: pertama, ke OLAP Cubes (SSAS) untuk analisis multidimensi oleh pihak berwenang

seperti CEO dan divisi keuangan; dan kedua, ke dashboard analitik, yang dapat divisualisasikan dan digunakan untuk membuat keputusan tentang produksi versus konsumsi energi, serta untuk memantau kinerja peralatan oleh tim data analyst, divisi produksi, dan pihak terkait lainnya.

Bagian 2 Arsitektur

1. Arsitektur Logikal

Arsitektur logikal Data Warehouse ElSusEnv dibangun berdasarkan skema konseptual yang telah dikembangkan dan diimplementasikan pada Misi Pertama dan Kedua. Struktur ini terdiri dari lima *fact table* utama: produksi energi, performa alat, efisiensi operasional, dampak lingkungan, dan kepatuhan regulasi. Kelima fakta tersebut dirancang dengan skema *star schema* yang terpisah secara fungsional namun terintegrasi secara semantik melalui *conformed dimensions*, yaitu waktu, lokasi, peralatan, dan kebijakan regulatif.

Setiap *fact table* beroperasi dengan granularitas harian dan terhubung langsung ke dimensi kontekstual yang menyimpan atribut relevan untuk pelaporan dan analitik keberlanjutan. Dimensi waktu mencakup atribut standar serta pelabelan kalender keberlanjutan berdasarkan musim dan tahun fiskal. Dimensi lokasi memuat informasi struktur fasilitas dan wilayah administratif yang telah diselaraskan pada tahap transformasi data. Dimensi alat dan regulasi terhubung ke beberapa fakta dengan tujuan memungkinkan kueri lintas proses seperti evaluasi performa alat terhadap hasil inspeksi atau nilai kepatuhan terhadap emisi.

Pendekatan *Slowly Changing Dimension Type 2* diterapkan pada atribut-atribut yang mengalami perubahan periodik dan berdampak pada analitik longitudinal. Implementasi SCD2 digunakan secara eksplisit pada status alat, klasifikasi proyek efisiensi, serta cakupan kebijakan regulasi lingkungan. Seluruh dimensi dibangun berdasarkan data yang tersedia pada sumber eksternal seperti Kaggle dan EIA, serta data internal yang telah melalui proses validasi di Misi Kedua.

Hubungan antar entitas dibentuk tidak hanya untuk memfasilitasi pelaporan, tetapi juga untuk memungkinkan evaluasi kausalitas dalam konteks lingkungan. Produksi energi yang tercatat dalam *fact_energy_production* dapat dianalisis bersama dengan tingkat downtime pada *fact_equipment_performance* dan nilai kepatuhan dari *fact_regulatory_compliance* menggunakan dimensi waktu dan lokasi yang konsisten. Dengan arsitektur ini, sistem mampu menjawab kebutuhan analitik multidimensi, lintas domain, dan historis yang diperlukan oleh ElSusEnv dalam mendukung operasional berbasis data dan prinsip keberlanjutan.

2. Arsitektur Fisik

Desain fisik Data Warehouse ElSusEnv difokuskan pada keandalan pemrosesan data skala besar, efisiensi penyimpanan, dan dukungan terhadap kebutuhan analitik harian maupun historis. Seluruh tabel disimpan dalam format *Apache Parquet* untuk memanfaatkan struktur berbasis kolom dan kemampuan kompresi tinggi. Penyimpanan dilakukan dalam file yang terpartisi berdasarkan *time_id* dan *site_id*, sesuai dengan struktur kueri utama yang telah diidentifikasi sejak tahap eksplorasi pada Misi Kedua.

Data dimuat secara harian melalui pipeline *extract-transform-load* yang dibangun menggunakan Python dan dijalankan terjadwal menggunakan Apache Airflow. Proses ini mencakup konversi satuan energi ke dalam MWh, suhu dalam Celsius, serta penyesuaian format waktu ke dalam struktur dimensi waktu terstandar. Pembersihan data dan validasi referensial juga dilakukan terhadap seluruh dimensi menggunakan kunci bisnis dan *lookup mapping* yang telah dibangun sebelumnya.

Indeks dibentuk secara terfokus pada kolom yang memiliki tingkat selektivitas tinggi dalam kueri pelaporan dan agregasi. Kolom seperti *equipment_id*, *emission_type_id*, dan *regulation_category_id* diprioritaskan karena terbukti digunakan secara berulang pada skenario analitik utama yang dievaluasi selama Misi Kedua. Strategi pembentukan indeks ini dilakukan adaptif berdasarkan analisis *query plan* dari hasil uji performa.

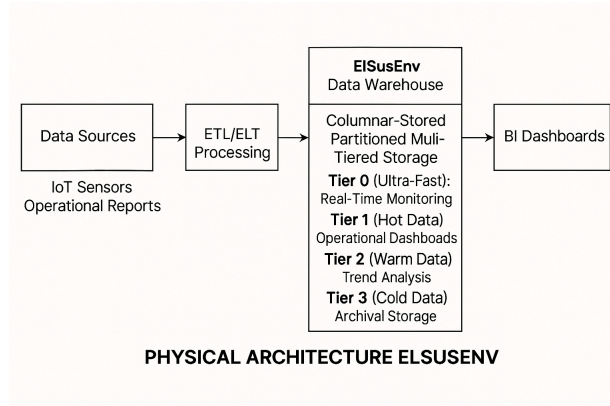
Struktur penyimpanan fisik diorganisasikan dalam tiga *tier* untuk mengelola data berdasarkan usia dan kebutuhan akses.

Tier 1 mencakup data aktif tiga bulan terakhir dan digunakan untuk dashboard harian seperti performa produksi energi, emisi harian, serta pemantauan alat. Data ini disimpan pada media latensi rendah untuk mendukung visualisasi interaktif melalui Power BI.

Tier 2 menyimpan data historis satu hingga dua tahun terakhir yang digunakan untuk evaluasi tren produksi energi tahunan, efektivitas proyek efisiensi, dan pelaporan triwulanan. Penyimpanan menggunakan media campuran dengan optimalisasi throughput dan penghematan ruang.

Tier 3 digunakan untuk menyimpan data historis lebih dari dua tahun yang mencakup hasil inspeksi terdahulu, pelaporan tahunan SDG, dan arsip proyek. Seluruh data pada tier ini disimpan dalam sistem *object storage* berbasis cloud dan tetap terhubung melalui metadata katalog yang memungkinkan akses historis saat dibutuhkan.

Dengan struktur partisi, format kolom, strategi indeks, dan tier penyimpanan ini, arsitektur fisik Data Warehouse ElSusEnv mampu menangani kebutuhan integrasi lintas domain, analitik keberlanjutan, serta pelaporan kepatuhan secara terstruktur, efisien, dan dapat diskalakan.



Gambar 2.1. Arsitektur Fisik Gudang Data ELSUS ENV Berbasis Penyimpanan Berlapis (Tiered Storage)

Bagian 3 ETL Pipeline

Pada Studi kasus ini Arsitektur ETL yang digunakan yakni *Three-Tier Architecture*. Desain Konseptual dari tugas misi sebelumnya yang dimana menggunakan *Database Northwind*. Dalam konteks perusahaan ELSUS ENV yang berfokus pada keberlanjutan dan efisiensi lingkungan, proses ETL (Extract, Transform, Load) dirancang untuk mentransformasikan data operasional ke dalam Data Warehouse berbasis pendekatan Data-Driven. Desain konseptual ETL ini diadaptasi dari struktur database Northwind, di mana entitas seperti Orders, Products, dan Employees disesuaikan menjadi Energy_Production, Equipment_Performance, dan Operational_Efficiency dalam konteks ELSUS ENV. Pada tahap Extract, data diambil dari sistem operasional seperti sistem pemantauan energi, sistem inspeksi peralatan, dan pencatatan pelaporan lingkungan. Data ini diekstrak dari berbagai sumber baik sensor IoT, log sistem, maupun entri manual.

Selanjutnya, pada tahap Transform, data mentah dibersihkan, dinormalisasi, dan dikonversi ke dalam bentuk yang sesuai dengan skema multidimensi yang telah dirancang. Contohnya, data emisi yang berasal dari berbagai perangkat dikonversi ke satuan kg CO₂e, data waktu diselaraskan ke format standar (YYYY-MM-DD), dan nilai status pelanggaran dikodekan menjadi level kepatuhan. Transformasi juga mencakup pembuatan fakta seperti Emission_Amount, Actual_Cost, dan Compliance_Score, serta pemetaan ke dimensi seperti Time, Site, dan Regulation. Terakhir, tahap Load akan memuat data yang telah dibersihkan ke dalam Data Warehouse ELSUS ENV, ke dalam tabel fakta seperti Environmental, Operational_Efficiency, dan Regulatory_Compliance, serta tabel dimensi pendukungnya. Pipeline ini memungkinkan ELSUS ENV melakukan analisis mendalam terhadap performa energi, emisi lingkungan, dan tingkat kepatuhan regulasi secara historis maupun real-time. Dengan desain ini, ELSUS ENV dapat mengambil keputusan strategis berbasis data yang akurat dan sesuai prinsip keberlanjutan.

Bagian 4 Alat yang digunakan

Ada beberapa *tools* atau alat yang digunakan pada tugas ketiga ini yaitu :

- *Microsoft SQL Server*
- *SQL Server Integration Services (SSIS)*
- *SQL Server Analysis Services (SSAS)*
- *SQL Server Reporting Services (SSRS)*

Microsoft SQL Server yang berfungsi sebagai sistem manajemen basis data relasional (RDBMS). *SQL Server* digunakan untuk menyimpan, mengelola, dan mengambil data secara paralel dengan bahasa pemrograman *SQL* (Structured Query Language). Komponen utama *SQL Server* yang digunakan adalah Database Engine untuk pemrosesan kueri dan manajemen basis data, serta Data Management System (SSMS) sebagai antarmuka pengguna grafis (GUI) dengan manajemen basis data.

SQL Server Integration Services (SSIS) sebagai alat integrasi, transform, load (ETL). *SSIS* digunakan untuk mengambil data dari berbagai sumber seperti lembar kerja Excel, CSV, atau basis data lainnya, lalu membersihkannya dan memuatnya ke gudang data. Proses ETL terdiri dari Aliran Kontrol dan Aliran Data untuk mengatur dan memindahkan data dari sumber ke tujuan secara efisien.

SQL Server Analysis Services (SSAS) digunakan dalam pengembangan model dan tabel berbasis OLAP (Online Analytical Processing) yang memfasilitasi analisis dan penelitian data. Pendekatan ini memungkinkan terciptanya kubus, hierarki, dan metrik yang membantu meningkatkan kinerja bisnis. Dalam studi ini, contoh tersebut digunakan karena cocok untuk alat analisis seperti Power BI.

SQL Server Reporting Services (SSRS) digunakan sebagai alat untuk membuat dan menampilkan laporan. *SSRS* menyediakan beberapa komponen yang berguna seperti Report Builder, Report Builder, dan Report Builder yang memungkinkan pengguna bisnis membuat laporan secara teknis dan visual. Laporan disusun dalam format bahasa definisi laporan (RDL) berbasis XML.

Dalam pendekatan modern, studi ini juga mempertimbangkan komputasi awan untuk optimasi sinaps dan optimasi sumber daya. Kedua teknologi ini digunakan sebagai bagian dari integrasi data dalam lingkungan cloud dan mendukung konsep arsitektur data modern seperti Data Lakehouse dan Medallion Architecture, yang mengatur data ke dalam kategori perunggu (data mentah), perak (data bersih), dan emas (data siap analisis).

Bagian 5 Query

Dalam upaya mendukung pengambilan keputusan untuk perusahaan ElSusEnv, perusahaan yang bergerak di energi dan sustainability disusunlah sebuah skrip SQL untuk membangun data warehouse yang terintegrasi. Implementasi skrip SQL dalam proses pembuatan data warehouse untuk merepresentasikan struktur data yang mendukung integrasi dan analisis. Skrip ini mencakup pembuatan tabel fakta dan dimensi, pendefinisian relasi antar tabel, serta penyusunan query analitik yang disesuaikan dengan kebutuhan analisis yang telah diidentifikasi, seperti monitoring produksi energi, evaluasi efisiensi operasional, pengukuran dampak lingkungan, dan audit kepatuhan terhadap regulasi. Berikut query pembuatan tabel fakta dan dimensi.

```
# 1. Time Dimension
CREATE TABLE dim_time (
    time_id INT PRIMARY KEY,
    date DATE,
    month INT,
    year INT,
    hours INT,
    minutes INT
);

# 2. Site Dimension
CREATE TABLE dim_site (
    site_id INT PRIMARY KEY,
    site_name VARCHAR(100),
    region VARCHAR(100),
    operational_status VARCHAR(50)
);

#3. Weather Dimension
CREATE TABLE dim_weather (
    weather_id INT PRIMARY KEY,
    temperature DECIMAL(5,2),
    humidity_percent DECIMAL(5,2),
    wind_speed_mps DECIMAL(5,2),
    rainfall_mm DECIMAL(5,2)
);

#4. Fuel Type Dimension
CREATE TABLE dim_fuel_type (
    fuel_type_id INT PRIMARY KEY,
    fuel_name VARCHAR(100),
    quantity DECIMAL(10,2)
);
```

#5. Equipment Dimension

```
CREATE TABLE dim_equipment (  
    equipment_id INT PRIMARY KEY,  
    equipment_name VARCHAR(100),  
    equipment_type VARCHAR(100),  
    manufacture VARCHAR(100),  
    capacity DECIMAL(10,2),  
    installation_date DATE,  
    last_maintenance_date DATE  
);
```

#6. Project Dimension

```
CREATE TABLE dim_project (  
    project_id INT PRIMARY KEY,  
    project_name VARCHAR(100),  
    project_type VARCHAR(50)  
);
```

#7. Emission Type Dimension

```
CREATE TABLE dim_emission_type (  
    emission_type_id INT PRIMARY KEY,  
    emission_scope VARCHAR(100),  
    source_type VARCHAR(100),  
    emission_unit VARCHAR(50)  
);
```

8. Regulation Dimension

```
CREATE TABLE dim_regulation (  
    regulation_id INT PRIMARY KEY,  
    enforcement_agency VARCHAR(100),  
    regulation_name VARCHAR(100),  
    compliance_level VARCHAR(50),  
    penalty_risk_level VARCHAR(50)  
);
```

9. Inspection Result Dimension

```
CREATE TABLE dim_inspection_result (  
    inspection_result_id INT PRIMARY KEY,  
    result_category VARCHAR(100),  
    severity_level VARCHAR(50),  
    action_required VARCHAR(100)  
);
```

#10. Fact Table: Energy Production

```
CREATE TABLE fact_energy_production (  
    production_id INT PRIMARY KEY,  
    time_id INT,
```



```
site_id INT,  
equipment_id INT,  
weather_id INT,  
fuel_type_id INT,  
total_production DECIMAL(12,2),  
cost_production DECIMAL(12,2),
```

FOREIGN KEY CONSTRAINTS

```
FOREIGN KEY (time_id) REFERENCES dim_time(time_id),  
FOREIGN KEY (site_id) REFERENCES dim_site(site_id),  
FOREIGN KEY (equipment_id) REFERENCES dim_equipment(equipment_id),  
FOREIGN KEY (weather_id) REFERENCES dim_weather(weather_id),  
FOREIGN KEY (fuel_type_id) REFERENCES dim_fuel_type(fuel_type_id)
```

);

#11. Fact Table: Equipment Performance

```
CREATE TABLE fact_equipment_performance (  
    ep_id INT PRIMARY KEY,  
    time_id INT,  
    site_id INT,  
    equipment_id INT,  
    fuel_type_id INT,  
    cost_maintenance DECIMAL(12,2),  
    fuel_consumption DECIMAL(12,2),  
    downtime_hours DECIMAL(10,2),
```

```
FOREIGN KEY (time_id) REFERENCES dim_time(time_id),  
FOREIGN KEY (site_id) REFERENCES dim_site(site_id),  
FOREIGN KEY (equipment_id) REFERENCES dim_equipment(equipment_id),  
FOREIGN KEY (fuel_type_id) REFERENCES dim_fuel_type(fuel_type_id)
```

);

#12. Fact Table: Operational Efficiency

```
CREATE TABLE fact_operational_efficiency (  
    oe_id INT PRIMARY KEY,  
    time_id INT,  
    site_id INT,  
    project_id INT,  
    actual_cost DECIMAL(12,2),  
    budget DECIMAL(12,2),  
    variance_cost DECIMAL(12,2),  
    duration INT, -- duration in days or hours
```

```
FOREIGN KEY (time_id) REFERENCES dim_time(time_id),  
FOREIGN KEY (site_id) REFERENCES dim_site(site_id),  
FOREIGN KEY (project_id) REFERENCES dim_project(project_id)
```

);

#13. Fact Table: Environmental

```

CREATE TABLE fact_environmental (
  env_id INT PRIMARY KEY,
  time_id INT,
  site_id INT,
  emission_type_id INT,
  regulation_id INT,
  carbon_emission DECIMAL(12,2),
  waste_generated DECIMAL(12,2),

  FOREIGN KEY (time_id) REFERENCES dim_time(time_id),
  FOREIGN KEY (site_id) REFERENCES dim_site(site_id),
  FOREIGN KEY (emission_type_id) REFERENCES
dim_emission_type(emission_type_id),
  FOREIGN KEY (regulation_id) REFERENCES dim_regulation(regulation_id)
);
#14. Fact Table: Regulatory Compliance
CREATE TABLE fact_regulatory_compliance (
  compliance_id INT PRIMARY KEY,
  time_id INT,
  site_id INT,
  regulation_id INT,
  inspection_result_id INT,
  inspection_count INT,
  violation_count INT,
  penalty_estimate DECIMAL(12,2),
  FOREIGN KEY (time_id) REFERENCES dim_time(time_id),
  FOREIGN KEY (site_id) REFERENCES dim_site(site_id),
  FOREIGN KEY (regulation_id) REFERENCES dim_regulation(regulation_id),
  FOREIGN KEY (inspection_result_id) REFERENCES
dim_inspection_result(inspection_result_id)
);

```

Berikut query analitik yang disesuaikan dengan kebutuhan analisis yang telah diidentifikasi sebelumnya:

1. Monitoring produksi energi

```

SELECT
  t.year, t.month,
  s.region,
  e.equipment_type,
  f.total_production,
  f.cost_production
FROM fact_energy_production f
JOIN dim_time t ON f.time_id = t.time_id

```

```
JOIN dim_site s ON f.site_id = s.site_id  
JOIN dim_equipment e ON f.equipment_id = e.equipment_id;
```

2. Evaluasi Efisiensi Operasional dan Biaya

```
SELECT  
    t.year, t.month,  
    s.site_name,  
    o.budget,  
    o.actual_cost,  
    o.variance_cost  
FROM fact_operational_efficiency o  
JOIN dim_time t ON o.time_id = t.time_id  
JOIN dim_site s ON o.site_id = s.site_id  
JOIN dim_project p ON o.project_id = p.project_id;
```

3. Evaluasi performa alat dan penyimpanan energi

```
SELECT  
    t.date,  
    e.equipment_name,  
    e.capacity,  
    ep.fuel_consumption,  
    ep.downtime_hours  
FROM fact_equipment_performance ep  
JOIN dim_time t ON ep.time_id = t.time_id  
JOIN dim_equipment e ON ep.equipment_id = e.equipment_id;
```

4. Audit emisi dan dampak lingkungan

```
SELECT  
    t.date,  
    s.region,  
    et.source_type AS emission_source,  
    f.carbon_emission,  
    f.waste_generated  
FROM fact_environmental f  
JOIN dim_time t ON f.time_id = t.time_id  
JOIN dim_site s ON f.site_id = s.site_id  
JOIN dim_emission_type et ON f.emission_type_id = et.emission_type_id;
```

5. Evaluasi Kepatuhan terhadap Regulasi

```
SELECT
    t.year,
    s.site_name,
    r.regulation_name,
    r.compliance_level,
    rc.inspection_count,
    rc.violation_count
FROM fact_regulatory_compliance rc
JOIN dim_time t ON rc.time_id = t.time_id
JOIN dim_site s ON rc.site_id = s.site_id
JOIN dim_regulation r ON rc.regulation_id = r.regulation_id;
```

```
CREATE NONCLUSTERED INDEX IDX_dim_time
ON dim_time ([date], [month], [year]);

CREATE NONCLUSTERED INDEX IDX_fact_energy_time ON fact_energy_production
(time_id);

CREATE NONCLUSTERED INDEX IDX_fact_energy_site ON fact_energy_production
(site_id);

CREATE FULLTEXT INDEX ON dim_regulation (regulation_name)
KEY INDEX PK_dim_regulation;
```