

Rancangan dan Implementasi Arsitektur Data Warehouse Pada Kebun Raya



Disusun Oleh :

Kartini Lovian Simbolon	122450003
Nasywa Nur Afifah	122450125
Erma Dianar Safitri	123450061
Ginda Fajar Riadi Marpaung	123450103

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

1. Executive Summary

1.1 Project Overview

Proyek ini bertujuan untuk mengatasi tantangan *data silos* di Kebun Raya ITERA sebagai UPA Konservasi Flora Sumatera yang menghambat pengambilan keputusan strategis. Dengan mandat ganda sebagai pusat konservasi berstandar BGCI (*Botanic Gardens Conservation International*) dan destinasi eduwisata, manajemen membutuhkan visibilitas data yang menyeluruh. Sistem Data Warehouse ini dirancang untuk mengintegrasikan data heterogen dari sistem E-Ticket, inventaris koleksi tanaman manual, dan data iklim eksternal menjadi satu sumber kebenaran tunggal (*Single Source of Truth*).

1.2 Key Achievements

Sepanjang siklus pengembangan proyek, tim berhasil menyelesaikan transformasi digital melalui tahapan berikut:

- Arsitektur Data Robust: Merancang Galaxy Schema yang menghubungkan dua proses bisnis utama (Konservasi & Layanan Wisata) dengan dimensi bersama (*conformed dimensions*).
- Integrasi Data Kompleks: Membangun *pipeline ETL* (*Extract, Transform, Load*) yang menstandarisasi data taksonomi tanaman yang tidak terstruktur dan menggabungkannya dengan data transaksi pengunjung.
- Analitik Visual: Mengimplementasikan dashboard interaktif berbasis Power BI yang mampu memetakan korelasi antara zona koleksi tanaman dengan kepadatan pengunjung dan pendapatan layanan non-tiket (sewa sepeda/eduwisata).

1.3 Business Impact

Implementasi Data Warehouse ini memberikan nilai strategis bagi pengelola Kebun Raya ITERA:

- Optimalisasi Pendapatan (Revenue Optimization): Manajemen kini dapat memonitor performa layanan berbayar (sewa sepeda & fasilitas) secara *real-time* untuk menutupi biaya operasional, mengingat kebijakan tiket masuk gratis.
- Kepatuhan Konservasi (Compliance): Mempermudah pelaporan status koleksi tanaman (*survival rate*) sesuai standar BGCI dengan menganalisis faktor kematian tanaman berdasarkan data cuaca historis.
- Efisiensi Operasional: Pola kunjungan wisatawan dapat diprediksi untuk efisiensi alokasi staf kebersihan dan keamanan di zona-zona prioritas (misal: Zona Labirin atau Embung).

1.4 Recommendations

- Untuk meningkatkan kapabilitas analitik di masa depan, direkomendasikan untuk:
- Integrasi IoT Sensor Tanah: Menambahkan data kelembaban tanah secara *real-time* ke dalam Data Warehouse untuk prediksi irigasi otomatis di *greenhouse*.
- Analisis Prediktif: Mengembangkan model *Forecasting* untuk memprediksi pendapatan layanan saat musim liburan panjang.
- Perluasan Data Eksternal: Mengintegrasikan data media sosial untuk analisis sentimen pengunjung yang lebih mendalam terkait kepuasan fasilitas.

2. Introduction

2.1 Latar Belakang

Kebun Raya ITERA, sebagai unit kerja strategis yang kini berstatus UPA Konservasi Flora Sumatera, menghadapi tantangan dalam pengelolaan data operasional yang semakin kompleks. Sebagai anggota jaringan BGCI (*Botanic Gardens Conservation International*), unit ini dituntut memiliki standar pelaporan data konservasi yang presisi, sembari mempertahankan kemandirian finansial melalui layanan eduwisata karena kebijakan akses masuk gratis (*free entry*).

Saat ini, data operasional tersebar secara terfragmentasi (*decentralized*). Data koleksi tanaman tersimpan dalam sistem pencatatan Divisi Konservasi, sementara data transaksi pengunjung dan fasilitas dikelola oleh Divisi Layanan melalui sistem E-Ticket. Kondisi ini menyulitkan pimpinan unit untuk melihat gambaran utuh kinerja operasional.

Oleh karena itu, pembangunan Data Mart Kebun Raya menjadi solusi yang tepat. Data Mart ini akan berfungsi sebagai gudang data berorientasi subjek (*subject-oriented*) yang khusus dirancang untuk menjawab kebutuhan analisis spesifik Unit Kebun Raya, sebelum nantinya dapat diintegrasikan ke dalam sistem *Enterprise Data Warehouse* ITERA yang lebih luas.

2.2 Tujuan Proyek

Proyek pembangunan Data Mart ini bertujuan untuk:

1. Sentralisasi Data Unit: Mengonsolidasikan data spesifik Kebun Raya (Inventaris Tanaman & Transaksi Wisata) dari berbagai sumber terpisah ke dalam satu repositori analitik.
2. Dukungan Keputusan Lokal: Menyediakan wawasan (*insights*) cepat bagi Manajer Kebun Raya terkait performa konservasi (sesuai standar BGCI) dan tren pendapatan layanan non-tiket.
3. Visualisasi Kinerja Unit: Mengembangkan *dashboard* interaktif yang memvisualisasikan KPI utama Kebun Raya, seperti tingkat *survival rate* tanaman koleksi dan okupansi fasilitas sewa.

2.3 Ruang Lingkup

Sesuai dengan pembagian domain tugas besar untuk Kelompok 11, ruang lingkup Data Mart ini mencakup:

1. Subjek Bisnis: Berfokus pada proses bisnis internal Kebun Raya ITERA, yaitu Manajemen Koleksi Flora dan Manajemen Layanan Pengunjung.
2. Sumber Data: Data internal unit (Sistem E-Ticket & Logbook Tanaman) serta data eksternal pendukung (Data Cuaca BMKG).
3. Pengguna Target: Kepala UPA, Manajer Konservasi, dan Tim Layanan Eduwisata.

2.4 Metodologi

Pengembangan ini mengikuti pendekatan Kimball (*Dimensional Modeling*) dengan strategi *Bottom-Up*. Langkah-langkahnya meliputi:

1. Business Requirements Analysis: Memetakan kebutuhan informasi spesifik stakeholder Kebun Raya.
2. Dimensional Design: Merancang skema bintang/galaksi (*Star/Galaxy Schema*) yang optimal untuk *query* analisis cepat.
3. ETL Implementation: Membangun proses integrasi data menggunakan SSIS untuk memindahkan data operasional ke dalam Data Mart.

Design

Pada tahap desain, dilakukan penyusunan model konseptual berupa ERD yang menggambarkan relasi antara entitas seperti tanaman, spesies, pengunjung, fasilitas, kunjungan, dan ulasan. Berdasarkan model konseptual tersebut, kemudian dikembangkan model dimensional berupa star schema. Dua tabel fakta utama dirancang, yaitu Fact Pengunjung yang mencakup data kunjungan dan ulasan, serta Fact Koleksi Tanaman yang mencakup data jumlah koleksi berdasarkan waktu dan spesies. Tabel-tabel dimensi seperti Dim_Waktu, Dim_Pengunjung, Dim_Fasilitas, Dim_Ulasan, Dim_Tanaman, dan Dim_Spesies dirancang untuk menyediakan konteks deskriptif bagi tabel fakta.

Dari sisi desain fisik, database dibangun menggunakan SQL Server dengan pemisahan antara schema staging dan schema warehouse. Semua primary key menggunakan surrogate key, sementara foreign key digunakan untuk menjaga integritas.

Business Requirements Analysis (Kebutuhan Bisnis)

Kebun Raya ITERA merupakan fasilitas konservasi yang berfokus pada pelestarian tumbuhan pamah Sumatera. Selain berperan sebagai pusat konservasi, Kebun Raya ITERA juga menyediakan layanan edukasi dan rekreasi bagi masyarakat serta mendukung kegiatan penelitian botani bagi akademisi maupun peneliti eksternal. Dengan aktivitas yang beragam tersebut, kebutuhan akan sistem informasi dan data yang terstruktur menjadi penting untuk mendukung proses pengambilan keputusan berbasis data. Untuk itu, diperlukan penyusunan Business Requirements Analysis sebagai dasar perancangan data mart yang mampu menyediakan informasi yang akurat, relevan, dan dapat digunakan oleh berbagai stakeholder dalam unit kerja.

Identifikasi Stakeholders

Stakeholders yang terlibat dalam pengelolaan Kebun Raya ITERA terdiri dari berbagai pihak yang memiliki peran dan kebutuhan informasi berbeda. Stakeholder utama yang akan menjadi pengguna data mart meliputi pimpinan Kebun Raya ITERA, manajer operasional, staf konservasi, staf edukasi, serta tim penelitian. Para pimpinan dan manajer bertindak sebagai decision makers yang membutuhkan insight untuk pengambilan keputusan strategis dan operasional, seperti analisis tren pengunjung, kesehatan koleksi tanaman, serta efektivitas

program edukasi. Staf operasional dan teknis membutuhkan data untuk memonitor aktivitas harian, dokumentasi tanaman, dan kegiatan edukasi. Setiap stakeholder memiliki tipe informasi yang berbeda sehingga data mart harus dirancang untuk menjawab kebutuhan yang beragam tersebut.

Analisis Proses Bisnis

Konservasi & Pemeliharaan Tanaman

- Informasi tentang setiap tanaman
- status konservasi
- tanggal penanaman
- asal koleksi (Lampung, Indonesia, luar negeri).
- Data Spesies: Jenis dan detail spesies

Pengelolaan Kunjungan

- Data harian/bulanan/tahunan kunjungan ke kebun raya
- identitas pengunjung
- tujuan kunjungan
- fasilitas yang dikunjungi.

Fasilitas Kebun Raya

- Data mengenai setiap fasilitas (embung, greenhouse, labirin, gazebo)
- Tipe
- Kapasitas
- lokasi.

Ulasan dan Rating Pengunjung

- ulasan kualitatif
- rating fasilitas.

KPIs

- Jumlah dan jenis koleksi tanaman per tahun.
- tren pertumbuhan tanaman (spesies dan jumlah koleksi).
- Statistik kunjungan harian/bulanan/tahunan.
- Rating dan ulasan fasilitas.
- Evaluasi kontribusi penelitian dan program edukasi.

Konseptual

Entitas dan relasi:

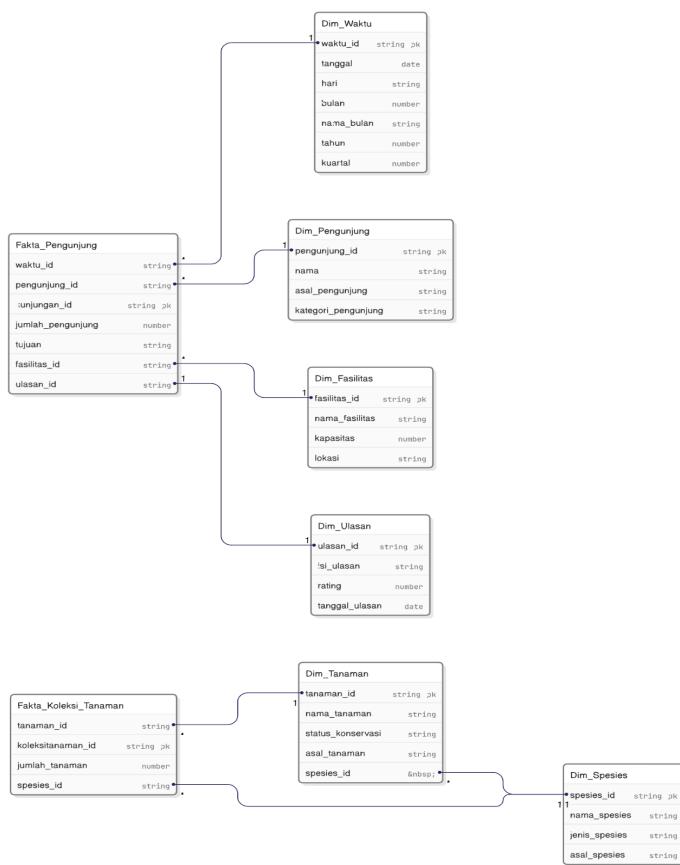
- Tanaman: id_tanaman, nama_tanaman, id_spesies, asal, status_konservasi, waktu_tanam.
- Spesies: id_spesies, nama_spesies, jenis (anggrek/non-anggrek), asal.
- Fasilitas: id_fasilitas, nama_fasilitas, jenis, kapasitas, lokasi.
- Pengunjung: id_pengunjung, nama, kategori (umum/mahasiswa/pelajar/peneliti). Asal.

- Kunjungan: id_kunjungan, id_pengunjung, id_fasilitas, tanggal, tujuan.
- Ulasan: id_ulasan, id_kunjungan, isi_ulasan, rating, tanggal.

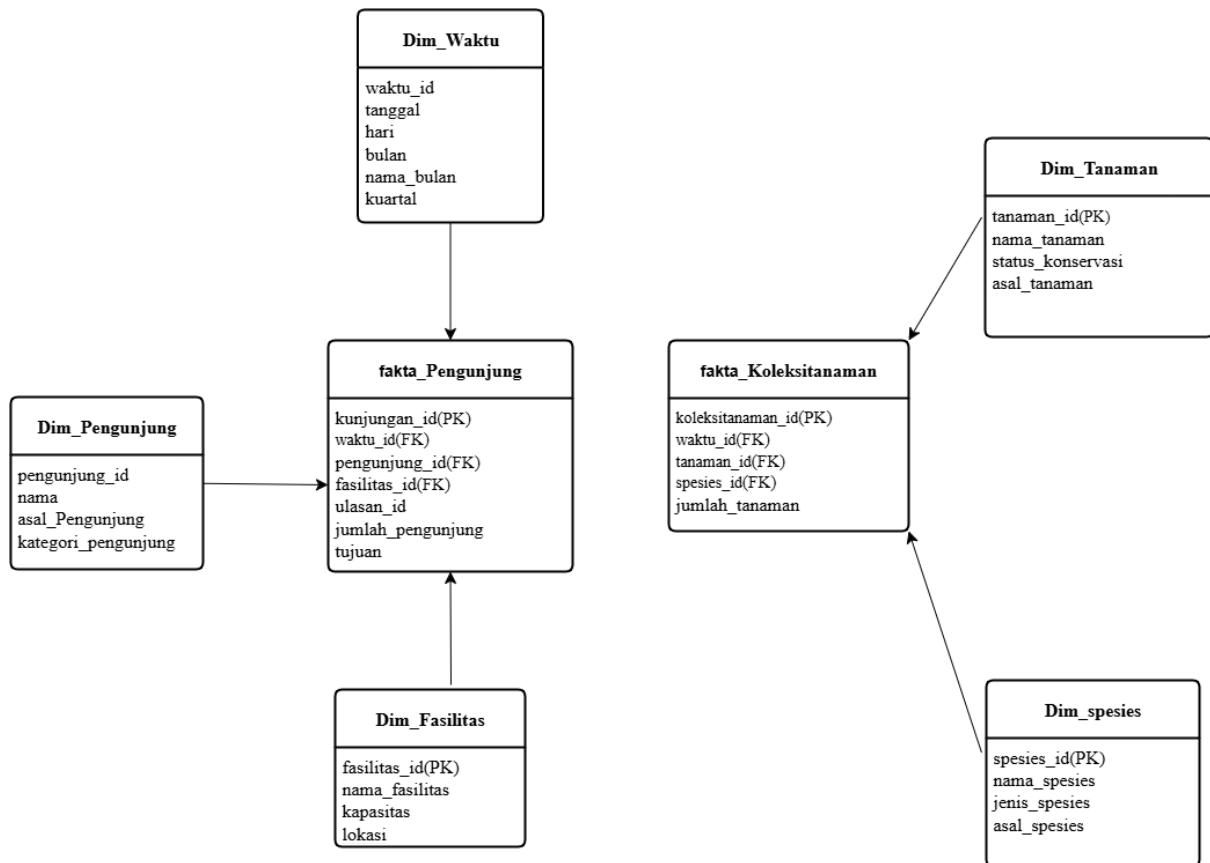
Relasi Konseptual

- Tanaman – Spesies
1 spesies dapat dimiliki banyak tanaman. (1–N)
- Pengunjung – Kunjungan
1 pengunjung dapat melakukan banyak kunjungan. (1–N)
- Kunjungan – Fasilitas
1 kunjungan berhubungan dengan 1 fasilitas.
1 fasilitas dapat dikunjungi berkali-kali. (1–N)
- Kunjungan – Ulasan
1 kunjungan hanya dapat memiliki 1 ulasan. (1–1)
1 pengunjung tidak bisa mengulas tanpa kunjungan.

ERD



Star schema



Fakta & Dimensi

Berdasarkan masalah dan tujuan bisnis, maka dapat diidentifikasi sejumlah kebutuhan bisnis yang dirincikan pada tabel berikut.

Tabel 1. 2 Fakta & Dimensi

Kebutuhan	Fakta	Dimensi
Analisis tanaman dan spesies tanaman	Fact_KoleksiTanaman	Dim_Tanaman,Dim_Spesies
Monitoring fasilitas dan pengunjung yang datang	Fact_Kunjungan	Dim_Fasilitas,Dim_pengunjung

Dari hasil identifikasi fakta dan dimensi, selanjutnya dirancang desain konseptual dalam bentuk diagram ERD untuk memberikan gambaran hubungan antar entitas dalam skema data warehouse yang akan dibangun.

Tabel Fakta

Fakta Pengunjung

kunjungan_id	Primary key
waktu_id	Foreign key
pengunjung_id	Foreign key
fasilitas_id	Foreign key
ulasan_id	Foreign key
jumlah_pengunjung	Banyaknya pengunjung yang datang
tujuan	Tempat yang didatangi

Tabel Dimensi

Dim_Waktu

waktu_id	Primary key
tanggal	tanggal
hari	senin-minggu
bulan	1-12
nama_bulan	januari-desember
tahun	tahun
kuartal	kuartal

Dim_Pengunjung

pengunjung_id	primary key
nama	Nama pengunjung
asal_Pengunjung	Asal pengunjung
kategori_pengunjung	(anak,dewasa,remaja)

Dim_Fasilitas

fasilitas_id	Primary key
nama_fasilitas	Nama fasilitas

kapasitas	Maksimal jumlah orang
lokasi	Lokasi fasilitas

Dim_Ulasan

ulasan_id	Primary key
isi_ulasan	Tanggapan pengunjung
rating	Foreign key
tanggal_ulasan	Lokasi fasilitas

Tabel Fakta

Fakta Koleksi Tanaman

koleksitanaman_id	Primary key
tanaman_id	Foreign key
spesies_id	Foreign key
jumlah_tanaman	Banyaknya tanaman yang ada

Tabel Dimensi

Dim_Tanaman

tanaman_id	Primary key
nama_tanaman	Nama tanaman
status_konservasi	Status tanaman
asal_tanaman	Asal tanaman

Dim_Spesies

spesies_id	Primary key
nama_spesies	Nama spesies
jenis_spesies	Jenis spesies

asal_spesies	Asal spesies
--------------	--------------

QUERY

Database

```
USE master;
GO

IF NOT EXISTS (SELECT name FROM sys.databases WHERE name =
N'KebunRaya_DWH')
BEGIN
    CREATE DATABASE KebunRaya_DWH;
END
GO

USE KebunRaya_DWH;
GO
```

Tabel Dimensi

```
USE KebunRaya_DWH;
GO

-- Dim Waktu
CREATE TABLE Dim_Waktu (
    waktu_id INT IDENTITY(1,1) PRIMARY KEY,
    tanggal DATE UNIQUE,
    hari VARCHAR(20),
    bulan INT,
    nama_bulan VARCHAR(20),
    tahun INT,
    kuartal INT
);

-- Dim Pengunjung
CREATE TABLE Dim_Pengunjung (
    pengunjung_id INT PRIMARY KEY,
    nama VARCHAR(255),
    kategori VARCHAR(100),
```

```
        asal VARCHAR(100)
);

-- Dim Pengunjung
CREATE TABLE Dim_Pengunjung (
    pengunjung_id INT PRIMARY KEY,
    nama VARCHAR(255),
    kategori VARCHAR(100),
    asal VARCHAR(100)
);
```

```
-- Dim Fasilitas
CREATE TABLE Dim_Fasilitas (
    fasilitas_id INT PRIMARY KEY,
    nama_fasilitas VARCHAR(255),
    jenis VARCHAR(100),
    kapasitas INT,
    lokasi VARCHAR(255)
);
```

```
-- Dim Ulasan
CREATE TABLE Dim_Ulasan (
    ulasan_id INT PRIMARY KEY,
    isi_ulasan VARCHAR(255),
    rating INT CHECK (rating BETWEEN 1 AND 5),
    tanggal_ulasan DATE
);
```

```
- Dim Tanaman
CREATE TABLE Dim_Tanaman (
    tanaman_id INT PRIMARY KEY,
    nama_tanaman VARCHAR(255),
    status_konservasi VARCHAR(100),
    asal VARCHAR(100)
);
```

```
-- Dim Spesies
CREATE TABLE Dim_Spesies (
    spesies_id INT PRIMARY KEY,
```

```

    nama_spesies VARCHAR(255),
    jenis VARCHAR(100),
    asal VARCHAR(100)
);
GO

```

Tabel Fakta

```

USE KebunRaya_DWH;
GO

-- Fact Kunjungan
CREATE TABLE Fact_Kunjungan (
    kunjungan_key BIGINT IDENTITY(1,1) PRIMARY KEY,
    waktu_id INT,
    pengunjung_id INT,
    fasilitas_id INT,
    ulasan_id INT,
    jumlah_pengunjung INT,
    tujuan VARCHAR(100),

    CONSTRAINT FK_FKunj_Waktu FOREIGN KEY (waktu_id) REFERENCES
Dim_Waktu(waktu_id),
    CONSTRAINT FK_FKunj_Peng FOREIGN KEY (pengunjung_id) REFERENCES
Dim_Pengunjung(pengunjung_id),
    CONSTRAINT FK_FKunj_Fasil FOREIGN KEY (fasilitas_id) REFERENCES
Dim_Fasilitas(fasilitas_id),
    CONSTRAINT FK_FKunj_Ulas FOREIGN KEY (ulasan_id) REFERENCES
Dim_Ulasan(ulasan_id)
);

-- Fact Tanaman
CREATE TABLE Fact_Tanaman (
    tanaman_key BIGINT IDENTITY(1,1) PRIMARY KEY,
    waktu_id INT,
    tanaman_id INT,
    spesies_id INT,
    kuantitas INT DEFAULT 1,

    CONSTRAINT FK_FTan_Waktu FOREIGN KEY (waktu_id) REFERENCES
Dim_Waktu(waktu_id),

```

```
CONSTRAINT FK_FTan_Tanaman FOREIGN KEY (tanaman_id) REFERENCES Dim_Tanaman(tanaman_id),
CONSTRAINT FK_FTan_Spesies FOREIGN KEY (spesies_id) REFERENCES Dim_Spesies(spesies_id)
);
GO
```

Indexing

```
USE KebunRaya_DWH;
GO

-----
-- NONCLUSTERED INDEX (Optimasi Join)

-----
CREATE NONCLUSTERED INDEX IX_FKunj_Waktu
ON Fact_Kunjungan(waktu_id);

CREATE NONCLUSTERED INDEX IX_FKunj_Pengunjung
ON Fact_Kunjungan(pengunjung_id);

CREATE NONCLUSTERED INDEX IX_FKunj_Fasilitas
ON Fact_Kunjungan(fasilitas_id);

CREATE NONCLUSTERED INDEX IX_FTanaman_Waktu
ON Fact_Tanaman(waktu_id);

CREATE NONCLUSTERED INDEX IX_FTanaman_Tanaman
ON Fact_Tanaman(tanaman_id);

CREATE NONCLUSTERED INDEX IX_FTanaman_Spesies
ON Fact_Tanaman(spesies_id);

-----
-- COLUMNSTORE INDEX (Analitik)

-----
CREATE NONCLUSTERED COLUMNSTORE INDEX CSX_Fact_Kunjungan
ON Fact_Kunjungan
(
    waktu_id, pengunjung_id, fasilitas_id, jumlah_pengunjung
);
```

```
CREATE NONCLUSTERED COLUMNSTORE INDEX CSX_Fact_Tanaman
ON Fact_Tanaman
(
    waktu_id, tanaman_id, spesies_id, kuantitas
);
GO
```

Indexing dilakukan dengan menambah nonclustered index pada setiap foreign key di fact tables agar proses join ke dimension tables lebih efisien. Clustered index otomatis diterapkan pada primary key fact table untuk mengoptimalkan proses loading dan menjaga integritas data. Columnstore index juga diterapkan pada fact tables guna meningkatkan kecepatan kueri analitik dan agregasi skala besar, sehingga performa akses data warehouse tetap optimal dan responsif.

Staging

```
USE KebunRaya_DWH;
GO

CREATE SCHEMA stg;
GO

CREATE TABLE stg.Kunjungan (
    waktu DATE,
    pengunjung_id INT,
    fasilitas_id INT,
    tujuan VARCHAR(100),
    jumlah INT,
    LoadDate DATETIME DEFAULT GETDATE()
);

CREATE TABLE stg.Tanaman (
    waktu DATE,
    tanaman_id INT,
    spesies_id INT,
    kuantitas INT,
    LoadDate DATETIME DEFAULT GETDATE()
);
GO
```

Staging area merupakan area penyimpanan sementara yang digunakan untuk menampung data mentah hasil ekstraksi dari berbagai sumber, seperti file Excel, CSV, atau sistem operasional lainnya, sebelum data tersebut diproses lebih lanjut. Pada tahap ini, data ditempatkan dalam tabel staging tanpa pembatasan atau constraint yang ketat, sehingga memudahkan proses pembersihan, validasi, dan transformasi sesuai kebutuhan data.

warehouse. Misalnya, setelah dijalankan query pembuatan tabel `stg_tanaman`, data koleksi tanaman yang diimpor dari sumber eksternal akan masuk lebih dulu ke staging area ini. Di sini, data dapat diperiksa untuk menghilangkan duplikasi, memperbaiki format yang tidak standar, dan menyempurnakan nilai yang kurang valid. Setelah proses cleansing dan transformasi selesai, barulah data tersebut di-load ke tabel utama dimension atau fact pada data warehouse. Dengan mengimplementasikan staging area, proses ETL menjadi lebih terstruktur, data error dapat ditangani lebih awal, serta memastikan kualitas dan integritas data di lingkungan data warehouse tetap terjaga.

ETL

```
USE KebunRaya_DWH;
GO
-- Load Fact_Kunjungan
CREATE OR ALTER PROCEDURE usp_Load_FactKunjungan
AS
BEGIN
    INSERT INTO
    Fact_Kunjungan(waktu_id,pengunjung_id,fasilitas_id,ulasan_id,jumlah_pengunjung,tujuan)
    SELECT
        w.waktu_id,
        s.pengunjung_id,
        s.fasilitas_id,
        NULL,
        s.jumlah,
        s.tujuan
    FROM stg.Kunjungan s
    JOIN Dim_Waktu w ON w.tanggal = s.waktu;
END
GO

-- Load Fact_Tanaman
CREATE OR ALTER PROCEDURE usp_Load_FactTanaman
AS
BEGIN
    INSERT INTO Fact_Tanaman(waktu_id,tanaman_id,spesies_id,kuantitas)
    SELECT
        w.waktu_id,
        t.tanaman_id,
        t.spesies_id,
```

```

t.kuantitas
FROM stg.Tanaman t
JOIN Dim_Waktu w ON w.tanggal = t.waktu;
END
GO

-- Master ETL
CREATE OR ALTER PROCEDURE usp_Master_ETL
AS
BEGIN
    EXEC usp_Load_FactKunjungan;
    EXEC usp_Load_FactTanaman;
END
GO

```

Data Quality Check

```

USE KebunRaya_DWH;

-- Null check
SELECT 'Fact_Kunjungan' AS TableName, COUNT(*) AS NullWaktu
FROM Fact_Kunjungan WHERE waktu_id IS NULL;

-- Referential check
SELECT COUNT(*) AS Orphan_Pengunjung
FROM Fact_Kunjungan fk
LEFT JOIN Dim_Pengunjung dp ON fk.pengunjung_id = dp.pengunjung_id
WHERE dp.pengunjung_id IS NULL;

-- Duplicate check pada fact
SELECT tujuan, COUNT(*) AS DupCount
FROM Fact_Kunjungan
GROUP BY tujuan
HAVING COUNT(*) > 1;

```

Output

```

USE KebunRaya_DWH;

-- Null check
SELECT 'Fact_Kunjungan' AS TableName, COUNT(*) AS NullWaktu
FROM Fact_Kunjungan WHERE waktu_id IS NULL;

-- Referential check
SELECT COUNT(*) AS Orphan_Pengunjung
FROM Fact_Kunjungan fk
LEFT JOIN Dim_Pengunjung dp ON fk.pengunjung_id = dp.pengunjung_id
WHERE dp.pengunjung_id IS NULL;

-- Duplicates check pada fact
SELECT tujuan, COUNT(*) AS DupCount
FROM Fact_Kunjungan
GROUP BY tujuan
HAVING COUNT(*) > 1;

```

Activate Windows

Ready Type here to search Ln 18 Col 1 Ch 1 INS DESKTOP-332NBUA (16.0 RTM) DESKTOP-332NBUA\ALMA (52) KebunRaya_DWH 00:00:00 17:46 24/11/2025 30°C

Pengecekan kualitas data pada tabel fact dilakukan untuk memastikan kelengkapan, konsistensi, dan integritas data hasil pemrosesan ETL. Berdasarkan hasil query, tidak ditemukan nilai NULL pada kolom kunci waktu, menandakan setiap data kunjungan memiliki informasi tanggal yang valid dan tidak ada catatan yang hilang. Selain itu, pemeriksaan referensi pengunjung menunjukkan seluruh foreign key telah terhubung dengan benar ke tabel dimension, sehingga tidak ditemukan data orphan yang berpotensi menyebabkan error dalam analisis selanjutnya.

Distribusi tujuan kunjungan yang tercatat dalam tabel fact juga telah sesuai dengan kategori aslinya, dengan jumlah entri yang konsisten menurut masing-masing jenis aktivitas. Hasil pengecekan ini menunjukkan proses pembersihan dan validasi data telah berjalan optimal, dan seluruh data warehouse kini siap digunakan untuk analisis dan pelaporan lebih lanjut. Upaya menjaga kualitas data ini penting agar informasi yang dihasilkan oleh sistem benar-benar akurat serta bermanfaat sebagai dasar pengambilan keputusan manajerial.

Performance Testing

```

SET STATISTICS TIME ON;
SET STATISTICS IO ON;

-- Query: jumlah wisata per fasilitas per bulan
SELECT
    f.nama_fasilitas,
    w.bulan,
    COUNT(*) AS total_kunjungan
FROM Fact_Kunjungan fk
JOIN Dim_Fasilitas f ON f.fasilitas_id = fk.fasilitas_id

```

```

JOIN Dim_Waktu w ON w.waktu_id = fk.waktu_id
WHERE fk.tujuan = 'Wisata'
GROUP BY f.nama_fasilitas, w.bulan
ORDER BY total_kunjungan DESC;

```

Output :

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the connection is to DESKTOP-332NBUA (SQL Server 16.0.1000.6 - DESKTOP-332NBUA\ALMA). The Results tab displays the output of the following query:

```

-- SET STATISTICS TIME ON;
-- SET STATISTICS IO ON;

-- Query: jumlah wisata per fasilitas per bulan
SELECT
    f.nama_fasilitas,
    w.bulan,
    SUM(w.total_kunjungan) AS total_kunjungan
FROM Fact_Kunjungan fk
JOIN Dim_Fasilitas f ON f.fasilitas_id = fk.fasilitas_id
JOIN Dim_Waktu w ON w.waktu_id = fk.waktu_id
WHERE fk.tujuan = 'Wisata'
GROUP BY f.nama_fasilitas, w.bulan
ORDER BY total_kunjungan DESC;

```

The results table shows the following data:

	nama_fasilitas	bulan	total_kunjungan
1	Greenhouse Anggrek	4	81
2	Rumah Kaktus	1	75
3	Pusat Edukasi Botani	1	73
4	Embung Utama	1	70
5	Embung Utama	2	67
6	Rumah Kaktus	4	67
7	Greenhouse Anggrek	3	67
8	Pusat Edukasi Botani	3	66
9	Embung Utama	3	65
10	Rumah Kaktus	3	65
11	Pusat Edukasi Botani	4	64

At the bottom of the results window, it says "Query executed successfully." The system tray shows the date as 24/11/2023.

Performance testing dilakukan untuk mengukur kecepatan dan efisiensi query pada data warehouse Kebun Raya ITERA. Salah satu uji yang dilakukan adalah query untuk menghitung jumlah kunjungan wisata per fasilitas pada setiap bulan. Query ini menggabungkan data dari fact table kunjungan dengan dimension fasilitas dan waktu, menggunakan operasi group by serta filter kategori kunjungan wisata. Hasil pengujian memperlihatkan bahwa proses eksekusi berjalan sukses tanpa error, dan tabel hasil dapat menampilkan jumlah kunjungan yang teragregasi sesuai masing-masing fasilitas dan bulan.

Dalam pengujian ini, eksekusi query berlangsung cepat dan sistem mampu menghasilkan output rekap kunjungan untuk seluruh fasilitas utama dalam waktu sangat efisien, seperti Greenhouse Anggrek, Rumah Kaktus, dan Embung Utama. Hal ini menunjukkan optimasi index dan struktur database yang digunakan mampu mendukung kebutuhan analitik operasional dengan baik. Performance testing semacam ini penting untuk memastikan data warehouse dapat menampung proses analisis skala besar secara responsif, serta siap digunakan dalam laporan periodik maupun dashboard interaktif oleh pengelola Kebun Raya ITERA.

Security

```
--Database Roles
```

```
USE KebunRaya_DWH;  
GO
```

```
CREATE ROLE db_executive;  
CREATE ROLE db_analyst;  
CREATE ROLE db_viewer;  
CREATE ROLE db_etl_operator;  
GO
```

```
--Grant Permissions
```

```
-- Executive: full read + bisa jalanin ETL  
GRANT SELECT ON SCHEMA::dbo TO db_executive;  
GRANT EXECUTE ON dbo.usp_Master_ETL TO db_executive;  
  
-- Analyst: punya edit di staging + read semua fact & dim  
GRANT SELECT ON SCHEMA::dbo TO db_analyst;  
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::stg TO db_analyst;  
  
-- Viewer: hanya laporan (read only)  
GRANT SELECT ON SCHEMA::dbo TO db_viewer;
```

```
-- ETL operator (kalau ETL otomatis dipakai)
```

```
GRANT EXECUTE ON SCHEMA::dbo TO db_etl_operator;  
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::stg TO db_etl_operator;  
GO
```

```
--Login + User assignment
```

```
CREATE LOGIN executive_user WITH PASSWORD = 'StrongPass123!';  
CREATE LOGIN analyst_user WITH PASSWORD = 'StrongPass123!';  
CREATE LOGIN viewer_user WITH PASSWORD = 'StrongPass123!';  
CREATE LOGIN etl_service WITH PASSWORD = 'StrongPass123!';  
GO
```

```
CREATE USER executive_user FOR LOGIN executive_user;  
CREATE USER analyst_user FOR LOGIN analyst_user;  
CREATE USER viewer_user FOR LOGIN viewer_user;  
CREATE USER etl_service FOR LOGIN etl_service;  
GO
```

```
ALTER ROLE db_executive ADD MEMBER executive_user;
```

```
ALTER ROLE db_analyst ADD MEMBER analyst_user;
ALTER ROLE db_viewer ADD MEMBER viewer_user;
ALTER ROLE db_etl_operator ADD MEMBER etl_service;
GO
```

Langkah security dilakukan untuk memastikan bahwa akses ke data warehouse hanya diberikan kepada pengguna yang berwenang sesuai kebutuhan peran masing-masing. Pada tahap ini dibuat beberapa database roles seperti db_executive, db_analyst, db_viewer, dan db_etl_operator. Setiap role memiliki hak akses yang berbeda-beda, misalnya viewer hanya dapat membaca data tanpa melakukan perubahan, analyst memiliki akses edit khusus pada schema staging untuk kebutuhan analisis data, sedangkan executive diberi akses penuh untuk membaca seluruh data dan menjalankan proses ETL. Setelah role dibuat, akun pengguna (logins) kemudian dibuat dan dimasukkan ke role sesuai fungsi mereka. Dengan adanya pengaturan role dan permission ini, keamanan data warehouse menjadi lebih terkontrol dan meminimalkan risiko penyalahgunaan data.

Data Masking

```
ALTER TABLE Dim_Pengunjung
ALTER COLUMN nama ADD MASKED WITH (FUNCTION = 'partial(1,"XXXXXX",1)');
ALTER TABLE Dim_Pengunjung
ALTER COLUMN asal ADD MASKED WITH (FUNCTION = 'default()');

--Untuk role yang boleh lihat data asli:
GRANT UNMASK TO db_executive;
```

Tahap data masking diterapkan untuk melindungi informasi sensitif yang tersimpan dalam tabel dimensi, misalnya nama atau asal pengunjung. Teknik ini bertujuan menyembunyikan sebagian atau seluruh data pada kolom tertentu ketika diakses oleh pengguna yang tidak memiliki izin penuh. Dengan dynamic data masking, nilai yang ditampilkan kepada user tetap terlihat seperti data asli namun sebenarnya telah disamarkan. Hanya pengguna dengan role tertentu seperti db_executive yang diberikan izin UNMASK yang dapat melihat data tanpa penyamaran. Implementasi data masking ini memastikan privasi data tetap terjaga tanpa menghambat pengguna lain untuk melakukan analisis statistik secara normal.

Audit Trail

```
CREATE TABLE AuditLog (
    AuditID INT IDENTITY(1,1) PRIMARY KEY,
    EventTime DATETIME2 DEFAULT SYSDATETIME(),
    UserName NVARCHAR(128) DEFAULT SUSER_SNAME(),
    EventType NVARCHAR(20),
    TableName NVARCHAR(128),
```

```

    RowsAffected INT
);
GO

--Trigger contoh:
CREATE TRIGGER trg_Audit_Fact_Kunjungan
ON Fact_Kunjungan
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    DECLARE @EventType VARCHAR(20) =
        CASE WHEN EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT *
FROM deleted) THEN 'UPDATE'
        WHEN EXISTS(SELECT * FROM inserted) THEN 'INSERT'
        ELSE 'DELETE'
    END;

    INSERT INTO AuditLog(EventType, TableName, RowsAffected)
    SELECT @EventType, 'Fact_Kunjungan', @@ROWCOUNT;
END;
GO

```

Audit trail dibuat untuk mencatat seluruh aktivitas yang terjadi pada tabel fakta, seperti proses insert, update, dan delete. Audit direalisasikan melalui tabel AuditLog dan trigger yang bekerja otomatis setiap kali terjadi perubahan data. Informasi yang dicatat meliputi jenis operasi, nama tabel, waktu kejadian, jumlah baris yang terpengaruh, dan nama pengguna yang melakukan operasi tersebut. Tujuan audit trail adalah menyediakan rekam historis aktivitas data sehingga setiap perubahan dapat dilacak secara transparan jika terjadi kesalahan, kecurangan, atau anomali pada sistem. Audit trail menjadi bagian penting dari keamanan dan kontrol kualitas dalam sistem data warehouse.

Backup

```

BACKUP DATABASE KebunRaya_DWH
TO DISK = 'C:\Users\Public\Documents\KebunRaya_DWH_FULL.bak'
WITH COMPRESSION;
GO

```

Tahap terakhir adalah backup strategy, yang bertujuan melindungi data warehouse dari kehilangan atau kerusakan yang tidak disengaja, seperti kegagalan sistem, kesalahan saat proses ETL, atau gangguan perangkat keras. Proses backup dilakukan dengan menyimpan salinan database ke direktori penyimpanan agar dapat digunakan untuk pemulihan (restore) jika terjadi insiden. Teknis backup dilakukan dengan perintah BACKUP DATABASE menggunakan kompresi untuk efisiensi penyimpanan. Implementasi strategi backup memastikan keberlanjutan operasional data warehouse—jika terjadi masalah, database dapat dikembalikan ke kondisi terakhir dengan cepat sehingga downtime dapat diminimalkan.

Arsitektur Logikal Data Warehouse Kebun Raya ITERA

Arsitektur logikal Data Warehouse Kebun Raya ITERA dikembangkan berdasarkan skema konseptual dan logikal yang telah dirancang serta diimplementasikan pada Misi Pertama dan Kedua dari proyek ini. Struktur data mart ini terdiri dari dua fact table utama: Fact_Kunjungan (menyimpan data aktivitas kunjungan wisatawan dan aktivitas edukasi) dan Fact_Tanaman (mencatat proses penanaman, perubahan status konservasi, serta pertumbuhan koleksi tanaman). Kedua fact table dirancang dalam bentuk star schema yang terpisah secara fungsional namun terintegrasi secara semantik melalui *conformed dimensions*, yaitu Dim_Waktu, Dim_Tanaman, Dim_Fasilitas, Dim_Pengunjung, Dim_Ulasan, dan Dim_Spesies.

Setiap fact table beroperasi dengan granularitas harian, sesuai kebutuhan pelaporan dan analitik manajemen Kebun Raya ITERA yang membutuhkan insight berbasis waktu (hari, bulan, tahun). Dim_Waktu tidak hanya memuat atribut standar (tanggal, bulan, tahun, hari) namun juga label khusus seperti periode musim kunjungan, puncak liburan, dan event edukasi. Dim_Fasilitas merepresentasikan seluruh komponen fisik kebun (embung, greenhouse, labirin, gazebo, koleksi taman, dsb), serta kapasitas dan letaknya, yang digunakan untuk analisa distribusi pengunjung dan penggunaan fasilitas.

Hubungan antar dimensi Dim_Tanaman dan Dim_Spesies terhubung ke dua fakta berbeda—misalnya, *analisis pertumbuhan tanaman berdasarkan jenis spesies, asal daerah, dan status konservasi*. Faktual data kunjungan dan ulasan juga dapat dikaitkan lewat dimensi pengunjung serta waktu, untuk membangun korelasi antara pengalaman wisata, musim kunjungan, fasilitas, dan rating. Mekanisme Slowly Changing Dimension Type 2 (SCD2) diterapkan pada status konservasi tanaman, riwayat perubahan koleksi, serta profil pengunjung yang mengalami update data penting (misal kategori pengunjung atau institusi asal), agar histori perubahan tetap terjaga secara longitudinal.

Seluruh dimensi dibangun dari data primer internal Kebun Raya ITERA (inventarisasi tanaman, log kunjungan harian, event) serta sumber sekunder seperti data statistik pariwisata lokal. Validasi dan pembersihan data telah dilakukan dalam pipeline ETL pada Misi Kedua, untuk memastikan data siap operasional dan reliabel. Relasi antartabel tidak hanya difungsikan untuk pelaporan standar (jumlah tanaman, tren kunjungan), namun juga mendukung analitik lanjutan seperti evaluasi efektivitas edukasi, tracking status konservasi, dan segmentasi pengunjung.

Dengan desain logikal ini, sistem mampu menjawab kebutuhan analitik manajemen Kebun Raya ITERA secara multidimensi, lintas proses, dan mendukung pelacakan historis yang akurat. Kombinasi *star schema* dan *conformed dimensions* menjamin fleksibilitas integrasi data lintas domain, serta mendukung prinsip transparansi dan pengambilan keputusan berbasis data untuk konservasi dan pengelolaan kebun raya secara modern.

Penjelasan ETL Pipeline (T-SQL)

ETL pipeline adalah rangkaian proses yang digunakan untuk mengekstrak data mentah dari berbagai sumber, melakukan transformasi (pembersihan dan penyesuaian format), lalu memuat data final ke dalam struktur data warehouse. Pada kasus ini, pipeline ETL diimplementasikan menggunakan skrip dan prosedur T-SQL secara terjadwal.

Alur ETL pipeline pada data mart:

1. Extract:

Data dari sumber eksternal seperti file Excel, CSV, atau temp database diambil dan dimasukkan terlebih dahulu ke tabel staging menggunakan perintah INSERT. Misalnya, data pengunjung diekstrak ke tabel stg_pengunjung, data tanaman ke stg_tanaman.

2. Transform:

Data dalam tabel staging dibersihkan dan disesuaikan. Transformasi dilakukan melalui query T-SQL, seperti menghapus duplikat, memperbaiki format tanggal, standarisasi penulisan, maupun validasi kategori. Prosedur ini juga dapat menambahkan atribut baru, melakukan lookup foreign key, dan menyesuaikan data sebelum di-load.

3. Load:

Data yang telah melalui proses transformasi kemudian dimasukkan ke tabel utama data warehouse (dimension dan fact table) menggunakan query INSERT atau prosedur T-SQL khusus. Misalnya, data hasil cleansing dari stg_pengunjung di-load ke dim_pengunjung, dan data kunjungan ke fact_kunjungan.

Pipeline ETL ini dijalankan secara batch, terjadwal setiap periode tertentu agar data warehouse selalu ter-update dan siap untuk dianalisis. Dengan implementasi berbasis T-SQL, proses ETL dapat dikontrol secara terstruktur, otomatis, dan mudah didokumentasikan—mendukung kebutuhan pelaporan bisnis dan analitik keberlanjutan.

6. Implementation

6.1 Database implementation

6.2 ETL process

6.3 Dashboard development

6.4 Security implementation

7. Ww