

**LAPORAN FINAL
PEMBANGUNAN DATA MART KEBUN RAYA**



Disusun Oleh :

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

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

EXECUTIVE SUMMARY

1.1 Project Overview

Proyek pembangunan Data Warehouse Kebun Raya bertujuan menyediakan infrastruktur data terpusat yang mampu mendukung analisis kunjungan, fasilitas wisata, konservasi tanaman, serta performa operasional kebun raya. Sistem ini dirancang menggunakan pendekatan dimensional modeling dengan skema star schema, mencakup fact table dan dimension table yang mewakili proses bisnis utama. Selain itu, dibangun pula mekanisme ETL, data quality check, indexing, security, audit trail, dan data masking untuk memastikan performa, keamanan, dan kualitas data yang optimal.

1.2 Key Achievements

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

1. Arsitektur Data Robust: Merancang Galaxy Schema yang menghubungkan dua proses bisnis utama (Konservasi & Layanan Wisata) dengan dimensi bersama (*conformed dimensions*).
2. Integrasi Data Kompleks: Membangun *pipeline ETL (Extract, Transform, Load)* yang menstandarisasi data taksonomi tanaman yang tidak terstruktur dan menggabungkannya dengan data transaksi pengunjung.
3. 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 Mart ini memberikan nilai strategis bagi pengelola Kebun Raya ITERA:

1. Meningkatkan kemampuan manajemen dalam menganalisis pola kunjungan, fasilitas populer
2. Memberikan informasi yang berguna bagi tim konservasi mengenai kuantitas dan distribusi spesies tanaman..
3. Meningkatkan efisiensi operasional dengan menyediakan satu sumber data yang konsisten dan mudah diakses untuk seluruh pemangku kepentingan.

1.4 Recommendations

1. Menambahkan mekanisme Slowly Changing Dimension (SCD) terutama untuk Dim_Pengunjung dan Dim_Fasilitas.
2. Memperluas data sumber dengan menambahkan IoT sensor, seperti jumlah pengunjung realtime di gerbang masuk.
3. Integrasi Sistem GIS (Geographic Information System), Kebun Raya dapat menambahkan koordinat geografis dan zonasi pada data tanaman dan fasilitas. Integrasi ini memungkinkan analisis geospasial
4. Implementasi Data Lake untuk Menyimpan Data Tidak Terstruktur.

BAB I

PENDAHULUAN

1.1 Background

Pembangunan Data Warehouse Kebun Raya dilatarbelakangi oleh kebutuhan akan sistem informasi yang dapat mengintegrasikan data dari berbagai aktivitas seperti kunjungan, pemanfaatan fasilitas, serta konservasi tanaman. Sebelum adanya data warehouse, informasi tersebut seringkali terpisah dan tidak konsisten sehingga menyulitkan analisis komprehensif. seperti 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.

1.2 Tujuan

Proyek pembangunan Data Mart ini bertujuan untuk:

1. Sentralisasi Data Unit: Mengkonsolidasikan 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.

1.3 Batasan Masalah

Untuk memastikan bahwa implementasi Data Warehouse dan Data Mart Kebun Raya berjalan terarah, terukur, serta dapat diselesaikan dalam waktu pengembangan yang ditentukan, maka ruang lingkup proyek ini dibatasi pada beberapa aspek berikut:

1. Domain Proyek yang Difokuskan pada dua proses bisnis yaitu proses kunjungan pengunjung ke fasilitas, dan proses konservasi tanaman yang mencatat kuantitas tanaman berdasarkan spesies
2. Desain data warehouse menggunakan pendekatan Dimensional Modeling Kimball (Bottom-Up) dengan fokus pada pembentukan Departmental Data Mart, yaitu Data Mart Kunjungan dan Data Mart Konservasi.

3. Data yang digunakan terbatas pada dataset pengunjung, fasilitas, tanaman, spesies, waktu,tidak mencakup pengolahan data keuangan, tiket, event, inventaris, ataupun data operasional lainnya

1.4 Methodology

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.

BAB II

Requirements Analysis

2.1 Business Requirements Analysis

Kebun Raya ITERA adalah fasilitas konservasi yang berfokus pada pelestarian tumbuhan pamah Sumatera sekaligus menyediakan layanan edukasi, rekreasi, dan penelitian botani. Aktivitas yang beragam menuntut adanya sistem informasi yang terstruktur untuk mendukung pengambilan keputusan berbasis data. Karena itu, diperlukan Business Requirements Analysis sebagai dasar perancangan data mart yang mampu menyediakan informasi akurat, relevan, dan bermanfaat bagi seluruh stakeholder.

2.1.1 Identifikasi Stakeholders

Stakeholder	Peran	Kebutuhan
Manajer Operasional	Pengambil keputusan	Analisis kunjungan & fasilitas
Tim Konservasi	Pemeliharaan tanaman	Data spesies & kuantitas tanaman
Marketing	Promosi wisata	Tren tujuan pengunjung

2.1.2 Analisis Proses Bisnis

Proses bisnis utama Kebun Raya yang memerlukan dukungan Data Mart adalah dua proses inti: Proses Kunjungan Pengunjung dan Proses Konservasi Tanaman. Kedua proses ini menghasilkan data operasional yang perlu dianalisis secara historis untuk mendukung pengambilan keputusan manajerial, efisiensi operasional, serta perencanaan konservasi jangka panjang.

komponen	Deskripsi
Nama Proses Bisnis	Proses Kunjungan Pengunjung
KPI Utama	Jumlah kunjungan harian/bulanan/tahunan Persentase tujuan kunjungan Tingkat pemanfaatan fasilitas Tren kunjungan musiman
Metrik Analitik	jumlah pengunjung jumlah kunjungan per fasilitas tujuan kunjungan tren waktu (hari, bulan, tahun)

Komponen	Deskripsi
Nama Proses Bisnis	Proses Konservasi Tanaman
KPI Utama	Jumlah tanaman per spesies Jumlah spesies dengan status konservasi tertentu
Metrik Analitik	kuantitas tanaman kuantitas berdasarkan status konservasi

2.1.3 Kebutuhan Analitik

Pertanyaan Bisnis	Jenis Laporan yang Dibutuhkan	Level Agregasi & Granularitas
Fasilitas mana yang paling ramai?	Laporan jumlah kunjungan per fasilitas	Agregasi per fasilitas & Satu baris per kunjungan per fasilitas per tanggal
Berapa jumlah pengunjung per kategori?	Laporan distribusi pengunjung berdasarkan kategori	Agregasi per kategori pengunjung & Satu baris per pengunjung per fasilitas per tanggal
Apa tujuan kunjungan terbesar dalam satu tahun?	Laporan persentase tujuan kunjungan (wisata, edukasi, penelitian) per tahun	Agregasi per tujuan per tahun & Satu baris per kunjungan per tanggal
Bagaimana distribusi spesies berdasarkan jenis?	Laporan distribusi spesies berdasarkan jenis (kategori ilmiah)	Agregasi per jenis spesies & Satu baris per spesies per tanggal
Status konservasi apa yang paling dominan?	Laporan jumlah tanaman berdasarkan status konservasi	Agregasi per status konservasi & Satu baris per tanaman per tanggal

2.2 Data Source Identification

2.2.1 Identifikasi Sumber Data

Jenis Sumber Data	Deskripsi
Database Operasional (OLTP)	Sistem operasional internal Kebun Raya yang mencatat aktivitas harian, seperti data kunjungan pengunjung, pencatatan fasilitas, dan sistem inventaris tanaman. Contoh: <i>tabel kunjungan harian, tabel</i>

	<i>fasilitas, tabel pencatatan tanaman, tabel spesies.</i> Data ini merupakan sumber utama untuk Fact_Kunjungan dan Fact_Tanaman.
Sistem Konservasi Tanaman	Basis data internal yang digunakan oleh Divisi Konservasi untuk mencatat kuantitas tanaman, status konservasi, identifikasi spesies, serta hasil observasi lapangan. Informasi ini menjadi sumber langsung untuk Dim_Tanaman dan Dim_Spesies.
File Excel/CSV (Data Lapangan & Observasi Manual)	Data tambahan yang sering dicatat secara manual oleh petugas lapangan, seperti daftar tanaman yang diamati, catatan kuantitas tahunan, atau data pengunjung per kegiatan tertentu. Data ini biasanya memerlukan pembersihan dan validasi sebelum masuk ke staging.
Sistem Informasi Fasilitas (Facility Management System)	Sistem internal yang mencatat informasi fasilitas seperti nama fasilitas, lokasi, kapasitas, dan status operasional. Data digunakan sebagai sumber Dim_Fasilitas dalam DW.

2.2.2 Data Source Analysis

Aspek	Analisis
Struktur dan Schema	Sumber data Kebun Raya memiliki struktur yang beragam. Data kunjungan pengunjung dari sistem operasional biasanya terstruktur dengan baik (OLTP), mencakup tanggal kunjungan, kategori pengunjung, dan fasilitas yang digunakan. Sementara itu, data konservasi tanaman berasal dari sistem pencatatan internal yang cenderung semi-terstruktur, terutama pada atribut taksonomi dan status konservasi yang kadang masih berupa teks bebas hasil observasi lapangan.
Volume dan Growth Rate	Volume data pengunjung cenderung tinggi dan meningkat seiring bertambahnya aktivitas wisata setiap tahun. Setiap kunjungan menghasilkan baris baru pada fakta_Pengunjung. Sebaliknya, data

	konservasi tanaman memiliki pertumbuhan lebih rendah tetapi bersifat incremental berdasarkan observasi berkala.
Kualitas Data	Data pengunjung dan fasilitas dari OLTP relatif konsisten, tetapi data manual (laporan Excel, catatan lapangan) sering mengandung nilai hilang, format tidak seragam, dan potensi duplikasi. Data tanaman juga rentan inkonsistensi, misalnya perbedaan penulisan nama ilmiah atau status konservasi yang tidak diperbarui.
Frekuensi Update Data	Data kunjungan diperbarui secara harian mengikuti aktivitas operasional (daily batch load). Data fasilitas jarang berubah, hanya diperbarui ketika ada perubahan aset atau kapasitas. Data konservasi diperbarui berdasarkan observasi berkala, biasanya mingguan atau bulanan.

2.2.3 Data Profiling

Data Profiling dilakukan untuk mengevaluasi kualitas, kelengkapan, dan konsistensi data sebelum dimuat ke dalam Data Mart Kebun Raya. Proses ini mencakup pemeriksaan terhadap seluruh sumber data, yaitu data kunjungan pengunjung, fasilitas wisata, tanaman koleksi, dan spesies, yang berasal dari sistem operasional OLTP maupun file tambahan seperti Excel/CSV. Secara umum, Data Profiling memastikan bahwa data yang dimuat ke staging, dimensi, dan fact table telah memenuhi standar kualitas yang dibutuhkan untuk analitik dan pelaporan. Pemeriksaan meliputi identifikasi nilai kosong pada kolom kritis, deteksi duplikasi pada data pengunjung, fasilitas, spesies, serta pencatatan tanaman, serta validasi referensial untuk menjamin setiap foreign key memiliki padanan yang sah pada tabel dimensi. Selain itu dilakukan validasi tipe data dan rentang nilai untuk memastikan kolom numerik seperti jumlah pengunjung, kapasitas fasilitas, dan jumlah tanaman berada pada batas yang logis. Proses ini juga mencakup standarisasi format teks misalnya penulisan nama spesies, status konservasi, kategori pengunjung, dan nama fasilitas serta penyelarasian struktur tanggal melalui Dim_Waktu. Seluruh anomali dicatat di staging error log, sementara hanya data yang telah lolos validasi yang dimuat ke Fact_Kunjungan dan Fact_KoleksiTanaman. Dengan demikian, Data Profiling memastikan bahwa Data Warehouse Kebun Raya berisi data yang bersih, valid, dan siap digunakan untuk analisis kunjungan, performa fasilitas, serta pemantauan konservasi tanaman secara akurat dan berkelanjutan.

2.3 Functional Requirements

Sistem Data Warehouse Kebun Raya harus mampu mengintegrasikan data dari berbagai sumber melalui proses ETL yang meliputi ekstraksi dari sistem operasional, pemuatan ke staging, pembersihan data melalui penanganan nilai kosong, penghapusan duplikasi, standarisasi format, serta validasi tipe data dan referensial sebelum data dimigrasikan ke dalam struktur dimensional. Data yang telah diproses harus disimpan ke dalam skema bintang yang terdiri dari dua tabel fakta—Fact_Kunjungan dan Fact_KoleksiTanaman—beserta tabel dimensi seperti Dim_Waktu, Dim_Pengunjung, Dim_Fasilitas, Dim_Tanaman, dan Dim_Spesies. Sistem wajib menjamin integritas relasional antara tabel fakta dan dimensi melalui pemetaan kunci yang tepat, sekaligus menyediakan mekanisme pemeriksaan kualitas data seperti *null check*, *duplicate check*, dan *referential check* sebagaimana telah diimplementasikan dalam kode SQL. Data Warehouse ini juga harus mampu mendukung berbagai kebutuhan analitik, termasuk analisis pola kunjungan, fasilitas dengan trafik tertinggi, distribusi kategori pengunjung, tren tujuan kunjungan, serta distribusi dan status konservasi spesies tanaman. Selain itu, sistem harus memastikan performa query yang optimal melalui strategi indexing yang sesuai dan menyediakan struktur data yang siap diakses dan divisualisasikan dalam dashboard analitik untuk mendukung pengambilan keputusan manajerial dan operasional Kebun Raya secara efektif.

2.4 Non-functional Requirements

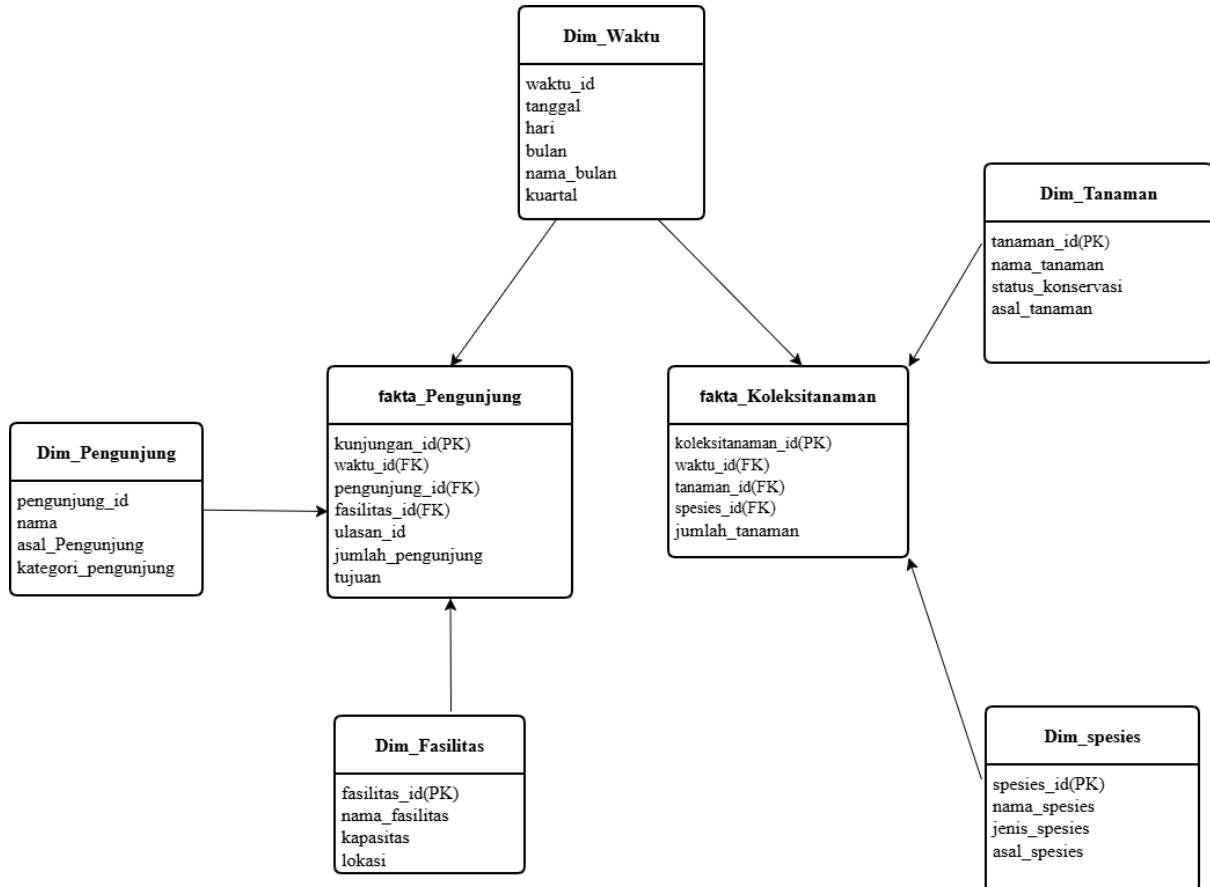
Data Warehouse Kebun Raya harus mampu memberikan performa pemrosesan yang cepat dan stabil, dengan ETL harian yang dapat diselesaikan dalam waktu yang efisien serta query analitik yang tetap responsif meskipun volume data terus bertambah. Sistem harus dapat diskalakan untuk mengakomodasi pertumbuhan data kunjungan dan konservasi serta penambahan domain analitik baru di masa depan, sekaligus menjaga reliabilitas dan ketersediaan dengan mekanisme backup, logging, dan penanganan kesalahan yang memadai. Kualitas data harus terjamin melalui pemeriksaan otomatis terhadap nilai hilang, duplikasi, integritas referensial, dan konsistensi format sebagaimana tercermin dalam proses ETL. Untuk menjaga keamanan, sistem harus menerapkan kontrol akses berbasis peran, pembatasan terhadap data sensitif, serta audit trail atas aktivitas pemrosesan data. Selain itu, Data Warehouse harus mudah dipelihara melalui struktur ETL yang modular, metadata yang terdokumentasi dengan baik, serta integrasi yang lancar dengan alat visualisasi seperti Power BI, sehingga analisis kunjungan dan konservasi dapat dilakukan secara efektif oleh seluruh pemangku kepentingan

BAB III

Perancangan

3.1 Conceptual model (ERD)

Dokumentasi ini menjelaskan setiap entitas (dimension dan fact), atribut yang dikandungnya, serta keterangan fungsional untuk mendukung analisis kunjungan dan konservasi tanaman, sesuai dengan kebutuhan analitik pada Business Requirements Analysis.



3.1.1 Identifikasi Entitas dan Atribut

Tabel ini menjelaskan Entitas dan atribut apa saja yang digunakan dalam Data Mart Kebun Raya

Entitas	Atribut	Keterangan
Dim_Waktu	waktu_id (PK), tanggal,hari,bulan,nam a_bulan,kuarter	Menyediakan konteks waktu untuk seluruh analisis. Digunakan dalam proses kunjungan dan konservasi untuk analisis tren musiman, tahunan, dan pertumbuhan koleksi tanaman.
Dim_Pengunjung	pengunjung_id(PK),	Menyimpan informasi

	nama, asal_pengunjung, kategori_pengunjung	deskriptif pengunjung. Mendukung segmentasi, perhitungan tren pengunjung
Dim_Fasilitas	Fasilitas_id(PK), nama_fasilitas, kapasitas, lokasi	Berisi informasi fasilitas yang digunakan dalam layanan wisata. Mendukung analisis okupansi, fasilitas paling ramai, dan pemanfaatan aset.
Dim_Tanaman	tanaman_id(PK), nama_tanaman, status_konservasi, asal_tanaman	Menyimpan data tanaman koleksi Kebun Raya. Mendukung analisis status konservasi dan pelestarian tumbuhan.
Dim_Spesies	spesies_id(PK), nama_spesies, jenis_spesies, asal_spesies	Dimensi taksonomi ilmiah yang memperkaya analisis biodiversitas. Menghubungkan data tanaman ke spesies ilmiah untuk klasifikasi dan analisis konservasi.
fakta_Pengunjung	kunjungan_id(PK), waktu_id(FK), pengunjung_id(FK), fasilitas_id(FK), jumlah _pengunjung, tujuan	Fact table utama untuk domain layanan wisata. Mencatat aktivitas kunjungan pengunjung ke fasilitas pada tanggal tertentu. Digunakan untuk KPI: jumlah pengunjung, fasilitas populer, tujuan kunjungan, tren tahunan.
fakta_KoleksiTanaman	koleksitanaman_id (PK),waktu_id(FK) tanaman_id(FK), spesies_id(FK),jumlah _tanaman	Fact table untuk domain konservasi. Mencatat jumlah tanaman berdasarkan spesies dan tanggal observasi. Mendukung KPI konservasi: jumlah tanaman per spesies, status konservasi dominan, dan pertumbuhan koleksi.

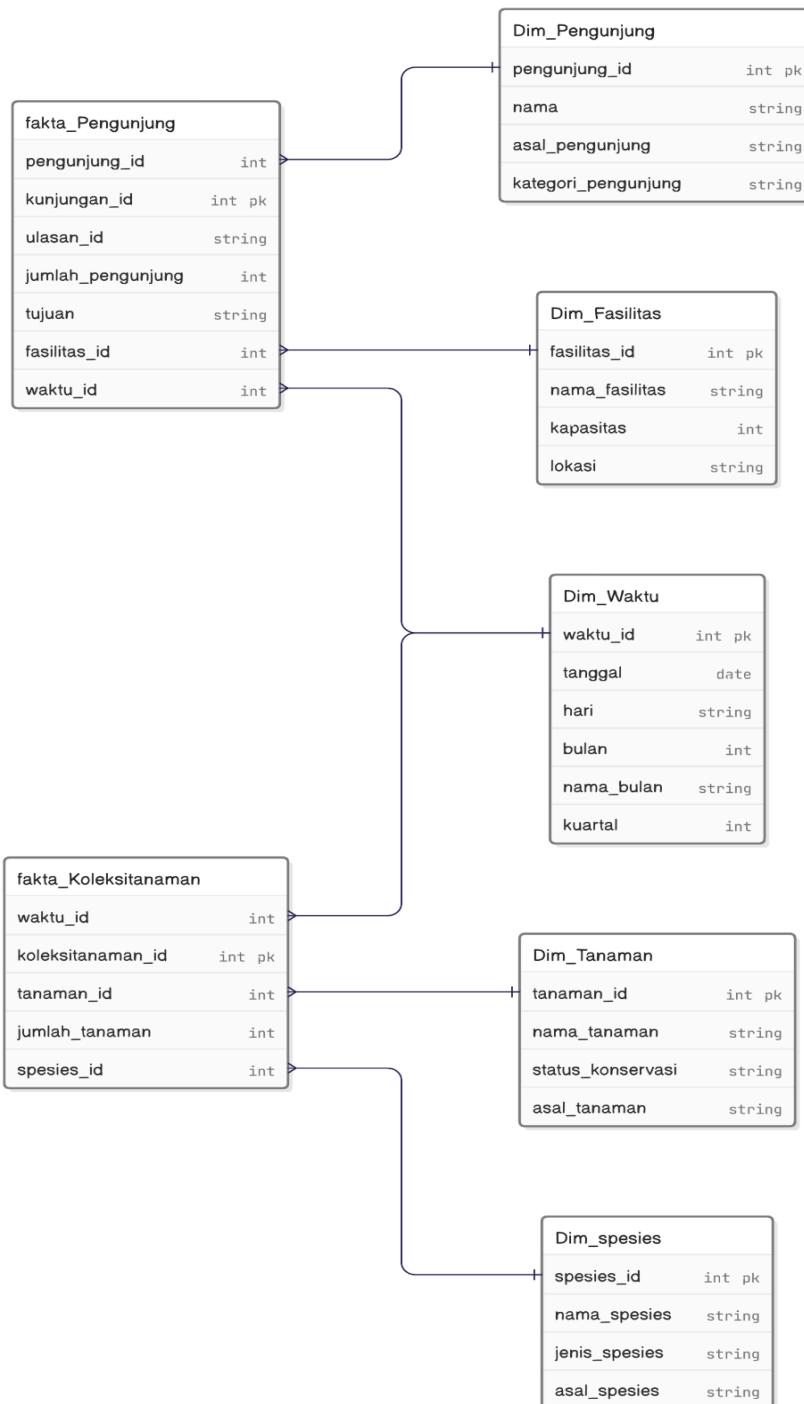
3.1.2 Definisi Relationships

Tabel ini menjelaskan hubungan antar entitas dalam Data Mart Kebun Raya, mencakup arah relasi, kardinalitas, dan fungsi hubungan tersebut dalam proses analitik.

Entitas	Kardinalitas	Keterangan
Dim_Waktu → fakta_Pengunjung	1 : Many	Setiap satu baris tanggal pada Dim_Waktu dapat memiliki banyak catatan kunjungan pada fakta_Pengunjung. Mendukung analisis tren waktu (harian, bulanan, tahunan).
Dim_Pengunjung → fakta_Pengunjung	1 : Many	Satu pengunjung dapat melakukan banyak kunjungan. Mendukung analisis segmentasi pengunjung dan perilaku kunjungan.
Dim_Fasilitas → fakta_Pengunjung	1 : Many	Setiap fasilitas dapat memiliki banyak kunjungan yang tercatat pada berbagai waktu. Relasi ini memungkinkan analisis fasilitas paling ramai dan kapasitas pemanfaatan.
Dim_Waktu → fakta_KoleksiTanaman	1 : Many	Setiap tanggal dapat memiliki banyak catatan koleksi tanaman berdasarkan observasi. Mendukung analisis perubahan jumlah tanaman dari waktu ke waktu.
Dim_Tanaman → fakta_KoleksiTanaman	1 : Many	Satu tanaman koleksi dapat dicatat berkali-kali pada waktu berbeda. Mendukung tracking kuantitas dan kondisi tanaman.
Dim_Spesies → fakta_KoleksiTanaman	1 : Many	Satu spesies dapat memiliki banyak catatan

kuantitas pada fact. Relasi ini memungkinkan analisis distribusi spesies berdasarkan jenis, asal, dan status konservasi.

3.2 Logical Design



3.2.1 Identifikasi Tabel Fact_Pengunjung

Aspek	Keterangan
Business Process	Proses Kunjungan Pengunjung ke fasilitas Kebun Raya. Proses ini mencatat aktivitas wisata, edukasi, dan penelitian yang dilakukan pengunjung pada hari tertentu.
Fact Table Name	fakta_Pengunjung
Grain (Level Detail)	Satu record merepresentasikan satu kejadian kunjungan pengunjung ke suatu fasilitas pada tanggal tertentu.
Measures (Metrik Numerik)	jumlah_pengunjung (jumlah individu dalam satu transaksi) frekuensi kunjungan (implicit count dari baris fact)
Klasifikasi Additivity	Additive (dapat dijumlahkan pada semua dimensi: waktu, fasilitas, kategori pengunjung).

Identifikasi Tabel Fact_KoleksiTanaman

Aspek	Keterangan
Business Process	Proses pencatatan koleksi tanaman oleh Divisi Konservasi, termasuk kuantitas tanaman berdasarkan spesies dan status konservasi pada waktu tertentu.
Fact Table Name	fakta_KoleksiTanaman
Grain (Level Detail)	Satu record merepresentasikan jumlah tanaman suatu spesies pada tanggal observasi tertentu.
Measures (Metrik Numerik)	jumlah_tanaman (kuantitas tanaman yang dicatat) - perubahan jumlah tanaman (dihitung dalam analitik)
Klasifikasi Additivity	Semi-Additive (additive across spesies & tanaman, tetapi tidak sepenuhnya additive across waktu karena data bersifat snapshot/stock value).

Identifikasi Dimension Tables

Dimensi	Atribut	Hierarki
Dim_Waktu	waktu_id (PK) tanggal hari bulan nama_bulan kuartal	Hari → Bulan → Kuartal → Tahun
Dim_Pengunjung	pengunjung_id (PK) nama asal_pengunjung kategori_pengunjung	Kategori Pengunjung → Asal Pengunjung → Pengunjung
Dim_Fasilitas	fasilitas_id (PK) nama_fasilitas kapasitas lokasi	Lokasi → Fasilitas
Dim_Tanaman	tanaman_id (PK) nama_tanaman status_konservasi asal_tanaman	Asal Tanaman → Tanaman → Status Konservasi
Dim_Spesies	spesies_id (PK) nama_spesies jenis_spesies asal_spesies	Jenis Spesies → Spesies Ilmiah → Asal Spesies

3.2.2 Design Star Schema

Desain Star Schema Data Warehouse Kebun Raya dibangun untuk merepresentasikan dua proses bisnis utama, yaitu kunjungan pengunjung dan konservasi tanaman, melalui dua fact table yang masing-masing dikelilingi oleh dimensi terkait. Fact fakta_Pengunjung mencatat setiap kejadian kunjungan pengunjung ke fasilitas pada tanggal tertentu dan terhubung dengan Dim_Waktu, Dim_Pengunjung, dan Dim_Fasilitas untuk memungkinkan analisis tren waktu, segmentasi pengunjung, serta evaluasi penggunaan fasilitas. Sementara itu, fact fakta_KoleksiTanaman menyimpan catatan jumlah tanaman berdasarkan spesies pada waktu observasi, yang dihubungkan dengan Dim_Waktu, Dim_Tanaman, dan Dim_Spesies, sehingga memungkinkan analisis konservasi seperti distribusi spesies, perubahan kuantitas tanaman, dan status konservasi. Dengan Dim_Waktu sebagai conformed dimension yang digunakan oleh kedua fact, Star Schema ini membentuk struktur analitik yang konsisten, sederhana, dan efisien, sekaligus mendukung kebutuhan pelaporan operasional maupun analisis strategis Kebun Raya ITERA secara menyeluruh.

3.2.3 Penentuan Surrogate Keys

Dalam pembangunan Data Warehouse Kebun Raya, seluruh tabel dimensi dirancang menggunakan Surrogate Keys untuk memastikan konsistensi identitas data, menjaga histori perubahan, serta menghindari ketergantungan pada struktur primary key dari sistem sumber. Penggunaan Surrogate Keys juga diperlukan karena sumber data Kebun Raya bersifat heterogen—berasal dari sistem kunjungan, sistem konservasi, file observasi manual, serta data fasilitas—sehingga diperlukan key yang seragam dan tidak berubah dari waktu ke waktu.

Pemanfaatan Surrogate Keys pada Dimensi

Setiap tabel dimensi diberi Surrogate Key berupa *integer auto-increment* yang berfungsi sebagai identifier tunggal. Kolom ini digunakan oleh fact table untuk membangun relasi yang stabil, terlepas dari perubahan yang terjadi pada sistem sumber. Contoh implementasi Surrogate Key pada dimensi:

- Dim_Waktu menggunakan *waktu_id* sebagai surrogate key untuk memetakan setiap tanggal unik.
- Dim_Pengunjung menggunakan *pengunjung_id* sebagai key yang merepresentasikan profile pengunjung.
- Dim_Fasilitas menggunakan *fasilitas_id* yang konsisten meskipun kapasitas atau nama fasilitas diperbarui.
- Dim_Tanaman menggunakan *tanaman_id* untuk memetakan tanaman koleksi.
- Dim_Spesies menggunakan *spesies_id* sebagai key untuk klasifikasi ilmiah setiap spesies.

Konvensi Penamaan Surrogate Key

Penamaan surrogate key dibuat sederhana dan konsisten, menggunakan format: *namaentitas_id*, misalnya:

- *waktu_id*
- *pengunjung_id*
- *fasilitas_id*
- *tanaman_id*
- *spesies_id*

Konvensi ini dipilih agar selaras dengan penamaan yang sudah digunakan dalam ETL, staging table, dan script yang sebelumnya kamu buat.

Surrogate Keys dalam Fact Table

Fact table menyimpan Foreign Key (FK) yang berasal dari surrogate key pada dimensi, sehingga semua relasi bersifat stabil, terukur, dan sesuai grain:

- fakta_Pengunjung menyimpan *waktu_id*, *pengunjung_id*, dan *fasilitas_id* sebagai FK.
- fakta_KoleksiTanaman menyimpan *waktu_id*, *tanaman_id*, dan *spesies_id* sebagai FK.

3.2.4 Data Dictionary

Table Dictionary fakta_Pengunjung

Kolom	Tipe Data	Keterangan
kunjungan_id	INT (PK)	Primary key untuk setiap catatan kunjungan.
waktu_id	INT (FK)	Foreign key menghubungkan ke Dim_Waktu.
pengunjung_id	INT (FK)	Foreign key menghubungkan ke Dim_Pengunjung.
fasilitas_id	INT (FK)	Foreign key menghubungkan ke Dim_Fasilitas.
ulasan_id	INT / NULL	ID ulasan jika tersedia (opsional).
jumlah_pengunjung	INT	Jumlah individu dalam satu transaksi kunjungan.
tujuan	VARCHAR	Tujuan: Wisata, Edukasi, Penelitian, Rombongan, dll.

Table Dictionary fakta_KoleksiTanaman

Kolom	Tipe Data	Keterangan
koleksitanaman_id	INT (PK)	Primary key untuk setiap catatan koleksi tanaman.
waktu_id	INT (FK)	Foreign key ke Dim_Waktu.
tanaman_id	INT (FK)	Foreign key ke Dim_Tanaman.
spesies_id	INT (FK)	Foreign key ke

		Dim_Spesies.
jumlah_tanaman	INT	Jumlah tanaman yang tercatat dalam observasi pada waktu tersebut.

Dimension Table Dictionary

Dim_Waktu

Kolom	Tipe Data	Keterangan
waktu_id	INT (Primary Key)	Surrogate key unik untuk setiap tanggal.
tanggal	DATE	Tanggal lengkap pencatatan kunjungan/observasi.
hari	VARCHAR(20)	Nama hari (Senin–Minggu).
bulan	INT	Nomor bulan (1–12).
nama_bulan	VARCHAR(20)	Nama bulan.
tahun	INT	Kuartal periode (1–4), digunakan untuk analisis musiman.

Dim_Pengunjung

Kolom	Tipe Data	Keterangan
pengunjung_id	INT (Primary Key)	Surrogate key unik untuk setiap pengunjung.
nama	VARCHAR(100)	Nama pengunjung (opsional, jika tersedia).
asal_pengunjung	VARCHAR(100)	Asal pengunjung (Lokal, Luar Kota, Provinsi, Nasional).
kategori_pengunjung	VARCHAR(100)	Kategori: Umum, Pelajar, Peneliti, Rombongan.

Dim_Fasilitas

Kolom	Tipe Data	Keterangan
fasilitas_id	INT (Primary Key)	Surrogate key untuk fasilitas wisata.
nama_fasilitas	VARCHAR(100)	Nama fasilitas di area Kebun Raya.
kapasitas	INT	Kapasitas maksimum fasilitas.
lokasi	VARCHAR(100)	Lokasi geografis fasilitas.

Dim_Tanaman

Kolom	Tipe Data	Keterangan
tanaman_id	INT (Primary Key)	Surrogate key untuk tanaman koleksi.
nama_tanaman	VARCHAR(100)	Nama umum tanaman.
status_konservasi	VARCHAR(100)	Status konservasi: Endangered (EN), Vulnerable (VU), Near Threatened (NT), Least Concern (LC).
asal_tanaman	VARCHAR(100)	Asal geografis tanaman (regional/nasional).

Dim_Spesies

Kolom	Tipe Data	Keterangan
spesies_id	INT (Primary Key)	Surrogate key untuk spesies ilmiah.
nama_spesies	VARCHAR(100)	Nama ilmiah spesies tanaman.
jenis_spesies	VARCHAR(100)	Kategori botani (pohon, perdu, palem, rumput, dll).

asal_spesies	VARCHAR(100)	Asal distribusi spesies.
--------------	--------------	--------------------------

3.3 Physical Database Design

3.3.1 Database Setup

```

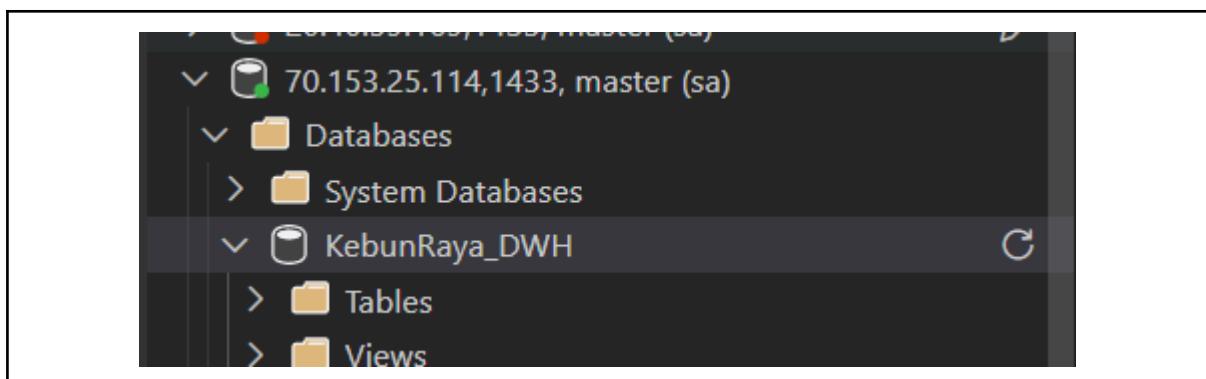
USE master;
GO

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

USE KebunRaya_DWH;
GO

```

Output



Langkah awal implementasi Physical Database Design adalah menciptakan wadah utama bagi Data Warehouse. Kode T-SQL ini memulai proses dengan mengarahkan sesi kerja ke database sistem master (USE master;) untuk mendapatkan hak administratif. Logika kondisional IF NOT EXISTS kemudian diterapkan untuk memastikan bahwa database KebunRaya_DWH belum ada. Jika database belum teridentifikasi, perintah CREATE DATABASE dijalankan. Pendekatan ini menjamin pembentukan database berlangsung secara aman dan terhindar dari error duplikasi. Terakhir, perintah USE KebunRaya_DWH; mengalihkan konteks kerja secara definitif, memastikan bahwa semua langkah selanjutnya, seperti pembuatan tabel dimensi dan fakta, akan ditempatkan secara terstruktur di dalam Data Mart Kebun Raya yang baru dibuat, yang kini siap untuk proses pemodelan dimensional dan ETL.

3.3.2 Create Dimension Tables

```
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
);

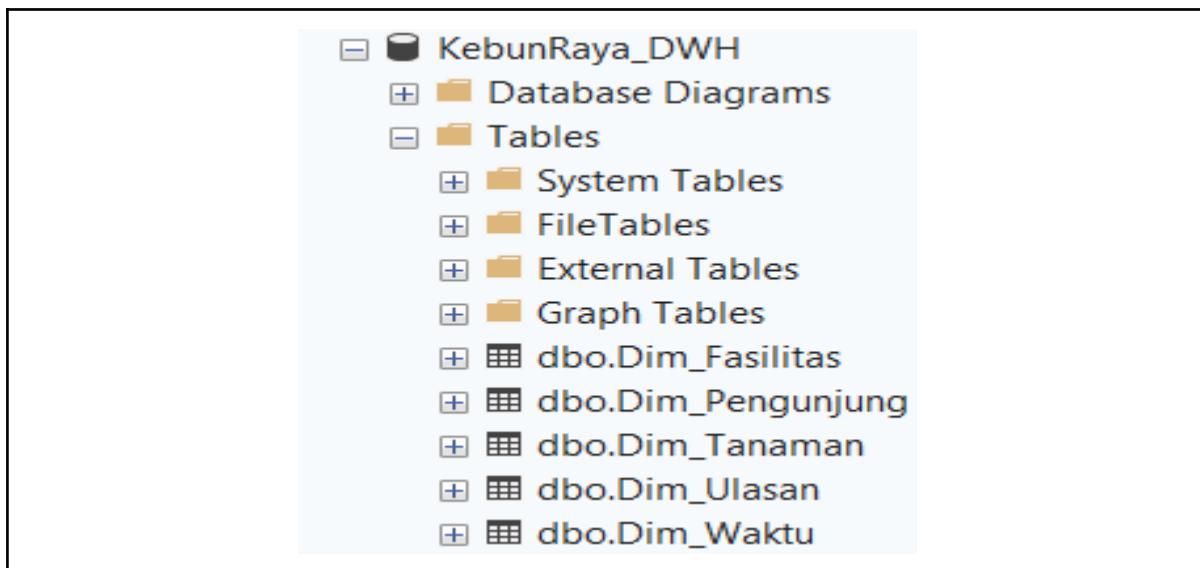
-- Dim Pengunjung
CREATE TABLE Dim_Pengunjung (
    pengunjung_id INT PRIMARY KEY,
    nama VARCHAR(255),
    kategori VARCHAR(100),
    asal VARCHAR(100)
);

-- Dim 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
);

CREATE TABLE dbo.Dim_Tanaman (
    SK_Tanaman INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    ID_Tanaman_Aslri VARCHAR(50) NOT NULL,
    Nama_Latin VARCHAR(255),
    Nama_Umum VARCHAR(255),
    Famili VARCHAR(100),
    Asal VARCHAR(100),
    Status_Konservasi VARCHAR(100),
    Lokasi_Zona VARCHAR(100)
);
GO
```

Output



Bagian skrip T-SQL ini memiliki fungsi krusial untuk mendirikan fondasi deskriptif bagi Data Mart, yaitu lima tabel dimensi yang akan mendukung slicing and dicing data pada Fact Table. Setiap tabel dimensi dirancang dengan kunci primer (Primary Key) untuk mengidentifikasi baris unik. Dim_Waktu dibuat sebagai conformed dimension menggunakan waktu_id sebagai Surrogate Key (IDENTITY(1,1)), yang merupakan standar untuk analisis tren musiman di kedua proses bisnis. Sementara itu, Dim_Tanaman juga menggunakan Surrogate Key otomatis (SK_Tanaman), dirancang untuk mengelola perubahan historis pada koleksi tanaman (Slowly Changing Dimension). Di sisi lain, Dim_Pengunjung, Dim_Fasilitas, dan Dim_Ulasan disiapkan untuk menerima data deskriptif dari sistem sumber. Secara kolektif, kelima dimensi ini menyediakan konteks yang diperlukan (siapa, apa, kapan, di mana) bagi metrik numerik yang tersimpan di Fact Table, sehingga memungkinkan pelaporan multi-dimensi yang cepat dan efisien.

3.3.3 Create Fact Tables

```
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 dbo.Fact_Tanaman (
    FactTanaman_ID BIGINT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    SK_Tanaman INT NOT NULL,
    Waktu_ID INT NOT NULL,
    Jumlah_Tanaman INT DEFAULT 1,
    Tinggi_cm DECIMAL(10,2),
    Diameter_cm DECIMAL(10,2),
    Kondisi_Kesehatan VARCHAR(50),

    CONSTRAINT FK_FactTanaman_DimTanaman FOREIGN KEY (SK_Tanaman)
        REFERENCES dbo.Dim_Tanaman (SK_Tanaman),

    CONSTRAINT FK_FactTanaman_DimWaktu FOREIGN KEY (Waktu_ID)
        REFERENCES dbo.Dim_Waktu (waktu_id)
);
GO

```

Output

The screenshot shows a file explorer interface for a database named 'KebunRaya_DWH'. The 'Tables' folder is expanded, displaying several tables: 'dbo.Dim_Fasilitas', 'dbo.Dim_Pengunjung', 'dbo.Dim_Tanaman', 'dbo.Dim_Ulasan', 'dbo.Dim_Waktu', 'dbo.Fact_Kunjungan', 'dbo.Fact_Tanaman', and 'Dropped Ledger Tables'. The 'dbo.Fact_Kunjungan' table is highlighted with a red rounded rectangle.

Bagian skrip T-SQL ini bertanggung jawab membangun inti dari Data Mart: dua Fact Table yang merepresentasikan metrik kuantitatif dari dua proses bisnis utama Kebun Raya, yaitu layanan wisata dan konservasi. Fact_Kunjungan mencatat metrik transaksi pengunjung, dengan kunjungan_key sebagai Primary Key IDENTITY dan kolom utama jumlah_pengunjung (metrik Additive). Tabel ini dihubungkan ke dimensi (waktu, pengunjung, fasilitas, dan ulasan) melalui Foreign Key (FK), memungkinkan analisis tren kunjungan harian, destinasi populer, dan segmentasi wisatawan. Sementara itu, Fact_Tanaman melayani domain konservasi, mencatat status koleksi dengan metrik Jumlah_Tanaman, Tinggi_cm, dan Diameter_cm; Fact Table ini terhubung ke Dim_Waktu dan Dim_Tanaman. Secara arsitektural, kedua Fact Table ini menggunakan Surrogate Key sebagai Primary Key (kunjungan_key dan FactTanaman_ID) untuk memastikan performa query yang cepat, dan secara kolektif membentuk Galaxy Schema dengan berbagai dimensi umum (Dim_Waktu), menyediakan pandangan analitik yang komprehensif atas kinerja operasional Kebun Raya.

3.3.4 Indexing Strategy

Clustered Index on Fact Table

```
USE KebunRaya_DWH;
GO

-----
-- 1. NONCLUSTERED INDEX (Optimasi Join / Filter)
-----

-- Index untuk Fact_Kunjungan (Tidak berubah)
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);

-- Index untuk Fact_Tanaman (Disediakan)
CREATE NONCLUSTERED INDEX IX_FTanaman_Waktu
ON Fact_Tanaman(Waktu_ID);

CREATE NONCLUSTERED INDEX IX_FTanaman_SK_Tanaman
ON Fact_Tanaman(SK_Tanaman); -- Menggunakan SK_Tanaman

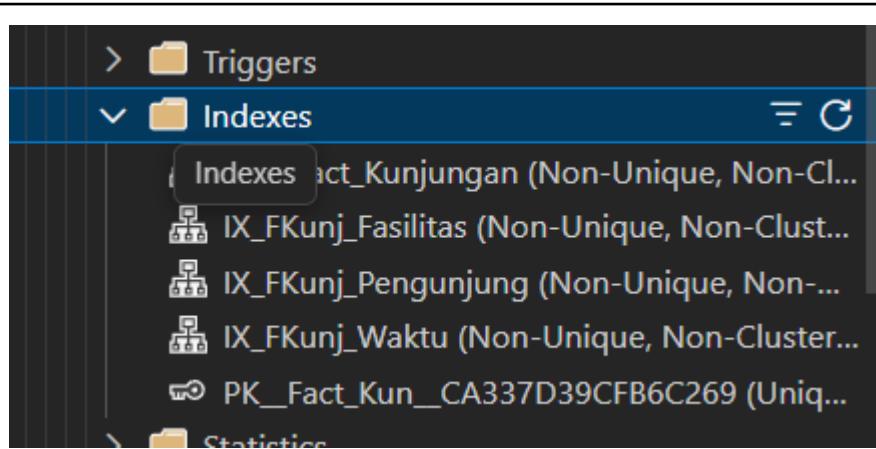
-- Catatan: Index Spesies dihapus karena kolomnya sudah tidak ada.

-----
```

```
-- 2. COLUMNSTORE INDEX (Optimasi Analitik / Agregasi)
-----
-- CS Index untuk Fact_Kunjungan
CREATE NONCLUSTERED COLUMNSTORE INDEX CSX_Fact_Kunjungan
ON Fact_Kunjungan
(
    waktu_id,
    pengunjung_id,
    fasilitas_id,
    jumlah_pengunjung,
    tujuan -- Ditambahkan agar filter berdasarkan tujuan lebih cepat
);

-- CS Index untuk Fact_Tanaman (Disesuaikan Kolom Baru)
CREATE NONCLUSTERED COLUMNSTORE INDEX CSX_Fact_Tanaman
ON Fact_Tanaman
(
    Waktu_ID,
    SK_Tanaman,
    Jumlah_Tanaman,      -- Pengganti kuantitas
    Tinggi_cm,           -- Kolom baru
    Diameter_cm,         -- Kolom baru
    Kondisi_Kesehatan   -- Kolom baru (untuk filter status
sehat/sakit)
);
GO
```

Output



Bagian skrip T-SQL ini mengimplementasikan strategi pengindeksan ganda (dual-indexing) pada Fact Table untuk mengoptimalkan performa query analitik dan join, yang merupakan bagian krusial dari Non-Functional Requirements. Strategi ini terdiri dari dua komponen utama. Pertama, Non-Clustered Index (IX_...) dibuat secara individual pada semua Foreign

Key (waktu_id, pengunjung_id, fasilitas_id, dan SK_Tanaman). Tujuan utama Index ini adalah mempercepat operasi JOIN antara Fact Table yang berisi jutaan rows dengan tabel dimensi, serta mempercepat operasi FILTER dasar. Kedua, Non-Clustered Columnstore Index (CSX_...) diterapkan pada kedua Fact Table. Jenis index ini secara khusus dirancang untuk mengoptimalkan beban kerja Data Warehouse (OLAP), karena ia menyimpan data dalam format kolom, menjadikannya sangat efisien dalam melakukan agregasi (seperti SUM atau COUNT) dan filter terhadap volume data yang sangat besar. Index ini mencakup kolom metrik (jumlah_pengunjung, Jumlah_Tanaman) dan Degenerate Dimension (tujuan), secara signifikan mempercepat laporan strategis Anda.

Non-Clustered Indexes

```
USE KebunRaya_DWH;
GO

SELECT
    t.name AS Nama_Tabel,
    i.name AS Nama_Index,
    i.type_desc AS Tipe_Index
FROM sys.indexes i
INNER JOIN sys.tables t ON i.object_id = t.object_id
WHERE t.name IN ('Fact_Tanaman', 'Fact_Kunjungan') -- Filter
tabel yang mau dicek
ORDER BY t.name, i.type_desc;
```

Output

	Nama_Tabel	Nama_Index	Tipe_Index
1	Fact_Kunjungan	PK_Fact_Kun_CA337D39CFB6C269	CLUSTERED
2	Fact_Kunjungan	IX_FKunj_Waktu	NONCLUSTERED
3	Fact_Kunjungan	IX_FKunj_Pengunjung	NONCLUSTERED
4	Fact_Kunjungan	IX_FKunj_Fasilitas	NONCLUSTERED
5	Fact_Kunjungan	CSX_Fact_Kunjungan	NONCLUSTERED COLUMNSTORE
6	Fact_Tanaman	PK_Fact_Tan_0D8FB0FFCC9B3197	CLUSTERED
7	Fact_Tanaman	IX_FTanaman_Waktu	NONCLUSTERED
8	Fact_Tanaman	IX_FTanaman_SK_Tanaman	NONCLUSTERED

Kode T-SQL ini bertujuan untuk melakukan verifikasi status implementasi Indexing Strategy dengan cara memeriksa katalog sistem (metadata) pada database KebunRaya_DWH. Query ini bekerja dengan menggabungkan (JOIN) dua tabel sistem krusial: sys.indexes (i), yang menyimpan informasi rinci tentang semua index yang ada, dan sys.tables (t), yang menyimpan daftar tabel. Hasil query difilter hanya untuk dua Fact Table utama

(Fact_Tanaman dan Fact_Kunjungan). Output dari query ini memberikan rincian penting seperti Nama_Tabel, Nama_Index (misalnya IX_FKunj_Waktu), dan Tipe_Index (type_desc, seperti NONCLUSTERED atau COLUMNSTORE). Informasi ini sangat penting pada tahap Validasi Performa karena mengkonfirmasi bahwa seluruh index yang dirancang telah berhasil terpasang di Fact Table, membuktikan bahwa strategi optimasi data telah diimplementasikan dengan benar sebelum pengujian stres.

3.3.5 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
```

Output

```

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 DATETIME
);

```

100 % Activate Windows

Messages
Commands completed successfully.
Completion time: 2025-11-24T17:44:00.4662436+07:00

100 % Query executed successfully.

DESKTOP-332NBUA (16.0 RTM) | DESKTOP-332NBUA\ALMA (55) | KebunRaya_DWH | 00:00:00 | 0 rows

Langkah ini merupakan inisialisasi Staging Area, yang merupakan lapisan perantara penting dalam arsitektur Data Warehouse. Proses dimulai dengan pembuatan skema terpisah *stg* (CREATE SCHEMA *stg*;) untuk secara logis mengisolasi tabel data mentah dari tabel dimensi dan fakta yang sudah bersih. Selanjutnya, skrip mendefinisikan dua *staging table* utama: *stg.Kunjungan* dan *stg.Tanaman*. Kedua tabel ini dirancang untuk menampung data mentah secara *as-is* (tanpa transformasi) dari sistem sumber, berfungsi sebagai *landing zone*. Kolom-kolomnya disesuaikan untuk mencerminkan skema data operasional. Khususnya, kolom *LoadDate* (DATETIME DEFAULT GETDATE()) ditambahkan secara otomatis pada kedua tabel. Kolom ini berfungsi sebagai metadata audit yang mencatat kapan tepatnya data berhasil dimuat ke *Staging Area*, yang sangat penting untuk pelacakan dan pemantauan proses ETL harian.

3.3.6 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_p
engunjung, 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

```

Output

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure under "DESKTOP-332NBUA (SQL Server 16.0.1000.6 - DESKTOP-332NBUA\ALMA)".
- Query Window:** Contains the T-SQL code from the previous section.
- Status Bar:** Displays "Query executed successfully." and other system information.

```

Object Explorer
Connect ▾ 08_ETL_Procedure...332NBUA\ALMA (57) 07_Create_Staging...332NBUA\ALMA (55) 05_Create_Indexes.sql - not connected
DESKTOP-332NBUA (SQL Server 16.0.1000.6 - DESKTOP-332NBUA\ALMA)
  Databases
    System Databases
    Database Snapshots
    AdventureWorksDW2022
    KebunRaya_DWH
  Security
  Server Objects
  Replication
  Always On High Availability
  Management
  Integration Services Catalogs
  SQL Server Agent (Agent XPs disabled)
  XEvent Profiler

08_ETL_Procedure...332NBUA\ALMA (57) 07_Create_Staging...332NBUA\ALMA (55) 05_Create_Indexes.sql - not connected
[08_ETL_Procedure...332NBUA\ALMA (57)] [07_Create_Staging...332NBUA\ALMA (55)] [05_Create_Indexes.sql - not connected]
[08_ETL_Procedure...332NBUA\ALMA (57)]
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

```

Skrip T-SQL ini mendefinisikan logika eksekusi untuk fase Load Data Mart menggunakan tiga Stored Procedure utama. Dua prosedur, `usp_Load_FactKunjungan` dan `usp_Load_FactTanaman`, adalah *loader* yang bertanggung jawab untuk memindahkan data dari Staging Area ke *Fact Table* akhir. Inti dari prosedur ini adalah operasi Key Lookup; prosedur tersebut menjalankan perintah SELECT dari tabel *staging* dan melakukan JOIN dengan dimensi yang sudah diisi (contohnya `Dim_Waktu`) untuk menukar *Natural Key* (tanggal sumber) menjadi Foreign Key yang sah (`waktu_id`), yang diperlukan untuk integritas *Star Schema*. Terakhir, prosedur `usp_Master_ETL` berfungsi sebagai orkestrator, yang tugasnya hanya menjadwalkan dan mengeksekusi urutan pemuatan data yang benar (EXEC `usp_Load_FactKunjungan`; dan EXEC `usp_Load_FactTanaman`). Prosedur *master* ini memastikan bahwa seluruh proses pemuatan data dapat dipicu secara andal dan otomatis melalui satu *job* tunggal yang dijadwalkan setiap hari.

3.4 Architecture diagram

Diagram ini menunjukkan dua Fact Table utama yang berbagi dimensi umum (`Dim_Waktu`) dan dimensi spesifik.

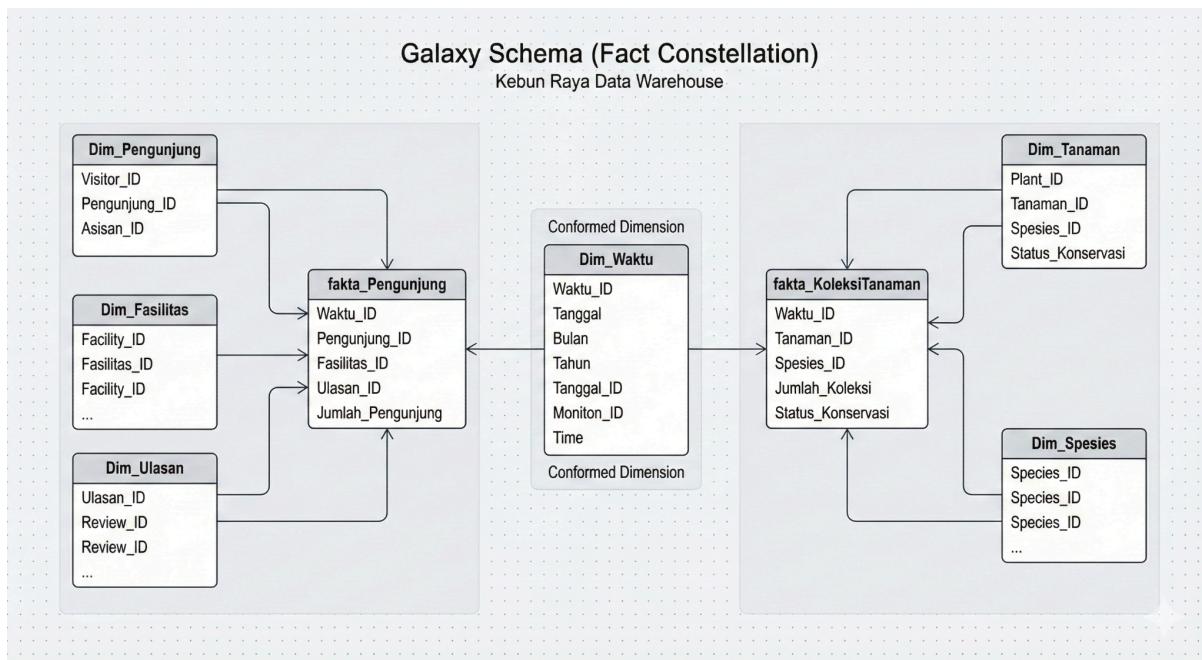


Diagram arsitektur Data Warehouse Kebun Raya mengimplementasikan model Galaxy Schema (Fact Constellation), yang merupakan pengembangan dari Star Schema untuk mendukung dua domain bisnis utama: Layanan Wisata dan Konservasi Tanaman.

Inti dari arsitektur ini adalah dua Fact Table independen:

`fakta_Pengunjung`: Mencatat metrik transaksi pengunjung, terhubung ke dimensi wisata (`Dim_Pengunjung`, `Dim_Fasilitas`, `Dim_Ulasan`).

`fakta_KoleksiTanaman`: Mencatat metrik konservasi tanaman, terhubung ke dimensi botani (`Dim_Tanaman` dan `Dim_Spesies`).

Kedua Fact Table tersebut berbagi dimensi umum (Dim_Waktu) yang berfungsi sebagai Conformed Dimension, memastikan bahwa analisis tren waktu (harian, bulanan, tahunan) konsisten di kedua domain. Model Galaxy ini memungkinkan granularitas detail yang tinggi untuk setiap proses bisnis sambil mempertahankan efisiensi dan integritas data, mendukung analisis silang antara data pengunjung dan data konservasi.

BAB IV

IMPLEMENTATION

4.1 Database Implementation

Pelaksanaan proses Extract, Transform, Load (ETL) dilakukan menggunakan Stored Procedure SQL yang telah dirancang pada tahap Misi 2 dan Misi 3. Mekanisme ini memastikan proses integrasi data berjalan terjadwal, konsisten, dan mudah dipantau. Proses ETL terdiri dari tiga tahap utama:

- Extract: Data diekstraksi dari file sumber (CSV/Excel) yang berisi informasi Kebun Raya ITERA seperti data Fasilitas, Pengunjung, Tanaman, Ulasan, Waktu dan Kunjungan.
- Transform: Pada tahap ini Transformasi mencakup pembersihan data dari nilai hilang, duplikasi, dan standarisasi format. Logika utama yang diterapkan di sini adalah pembuatan Surrogate Keys (seperti waktu_id atau tanaman_id) pada semua tabel dimensi, yang memutus ketergantungan pada kunci sistem sumber dan menjamin identitas data yang stabil. Selain itu, sistem harus menerapkan mekanisme Slowly Changing Dimension Type 2 (SCD2) pada atribut yang membutuhkan riwayat perubahan, seperti status konservasi tanaman dan profil pengunjung, untuk menjaga histori data. Setelah proses transformasi, validasi data, termasuk *null check* dan *referential check*, dilakukan sebelum data dimuat.
- Load: Tahap terakhir di mana data yang telah bersih dan bertransformasi dimuat ke dalam skema produksi dbo. Proses pemuatian harus dilakukan dengan urutan yang tepat: tabel dimensi dimuat terlebih dahulu, diikuti oleh tabel fakta (fakta_Pengunjung dan fakta_KoleksiTanaman).

4.1.1 Schema Creation

Skema pada database ini dirancang dengan menggunakan schema dbo (Production) untuk tabel dimensi dan fakta yang telah bersih dan terstruktur.

4.1.2 Partitioning & Indexing

- Partitioning function : Partitioning dilakukan berdasarkan dimensi waktu (Tanggal/Waktu) untuk mempermudah manajemen data dan efisiensi kueri analitik (partition elimination).
- Indexing : Index Non-Clustered (NC Index): Diterapkan pada seluruh kolom *Foreign Key* (FK) di tabel fakta. Index ini berfungsi untuk mempercepat operasi JOIN antar tabel fakta dan dimensi. Columnstore Index (CS Index): Diterapkan pada kedua tabel fakta (Fact_Kunjungan dan Fact_Tanaman). Index ini mengompresi data secara kolumnar dan secara signifikan mempercepat kueri agregasi skala besar.

4.1.2 Control Flow Logic

Alur kendali (*Control Flow*) pada proses ETL di Data Mart Kebun Raya diatur untuk menjalankan seluruh prosedur secara berurutan dan terkoordinasi, memastikan dimensi dimuat sebelum fakta. Proses ini mencakup empat tahap utama:

- ETL Dimensi
Tahap ini dilakukan melalui serangkaian prosedur yang bertujuan memuat dimensi yang bersifat statis atau jarang berubah. Setiap prosedur melakukan validasi, menambahkan data baru yang belum tersedia (*insert-if-not-exists*), serta memperbarui data secara inkremental. Prosedur ini mencakup pemuatan dimensi waktu, fasilitas, dan spesies, yaitu Dim_Waktu, Dim_Fasilitas, dan Dim_Spesies. Dim_Waktu sangat krusial karena ia berfungsi sebagai *conformed dimension* yang digunakan oleh kedua tabel fakta
- ETL Fact Table
ETL Fact Table dilakukan melalui prosedur pemuatan fakta yang bertugas memetakan *Foreign Key* (FK) dari fakta ke *Surrogate Key* (SK) pada dimensi yang telah dimuat sebelumnya. Proses ini mencakup pemuatan fakta_Pengunjung dan fakta_KoleksiTanaman. Prosedur ini bertanggung jawab melakukan validasi terhadap duplikasi *record* dan memasukkan berbagai *measures* (metrik numerik) seperti jumlah_pengunjung dan jumlah_tanaman.
- Finalization Step
merupakan tahap penutup untuk menjaga performa. Aktivitas utama meliputi penyegaran statistik, pencatatan aktivitas ke tabel *monitoring*, dan implementasi strategi pengindeksan, yaitu membangun Non-Clustered Index pada kolom FK dan Columnstore Index pada tabel fakta untuk mempercepat agregasi skala besar.

4.1.3 Data Flow Logic

Logika aliran data (Data Flow Logic) ini memetakan bagaimana data operasional Kebun Raya, yang terfragmentasi di berbagai sumber, diproses dan diselaraskan ke dalam struktur *Galaxy Schema* yang optimal untuk analitik.

- Extraction
Fase ini merupakan titik awal di mana data ditarik dari berbagai sumber internal dan eksternal. Sumber data mencakup Sistem E-Ticket & Layanan dan Inventaris Tanaman internal (termasuk logbook manual/Excel). Pada tahap ini, data dari file sumber disimpan tanpa perubahan struktural awal, menyediakan data mentah (*as-is*) sebagai dasar untuk proses pembersihan dan transformasi berikutnya.
- Transformation
Tahap transformasi berfungsi sebagai pusat pengolahan data untuk sanitasi dan penyeragaman. Proses ini sangat vital karena data sumber memiliki struktur yang beragam. Transformasi mencakup pembersihan duplikat, penanganan nilai kosong (*missing values*), dan penyelarasan format antar atribut. Secara spesifik, dilakukan standarisasi taksonomi tanaman dan format teks (misalnya nama spesies atau status konservasi). Selain itu, dilakukan pembuatan *Surrogate Key* (fasilitas_id, tanaman_id, dll.) yang bersifat *integer auto-increment* untuk semua tabel dimensi.

- Validation

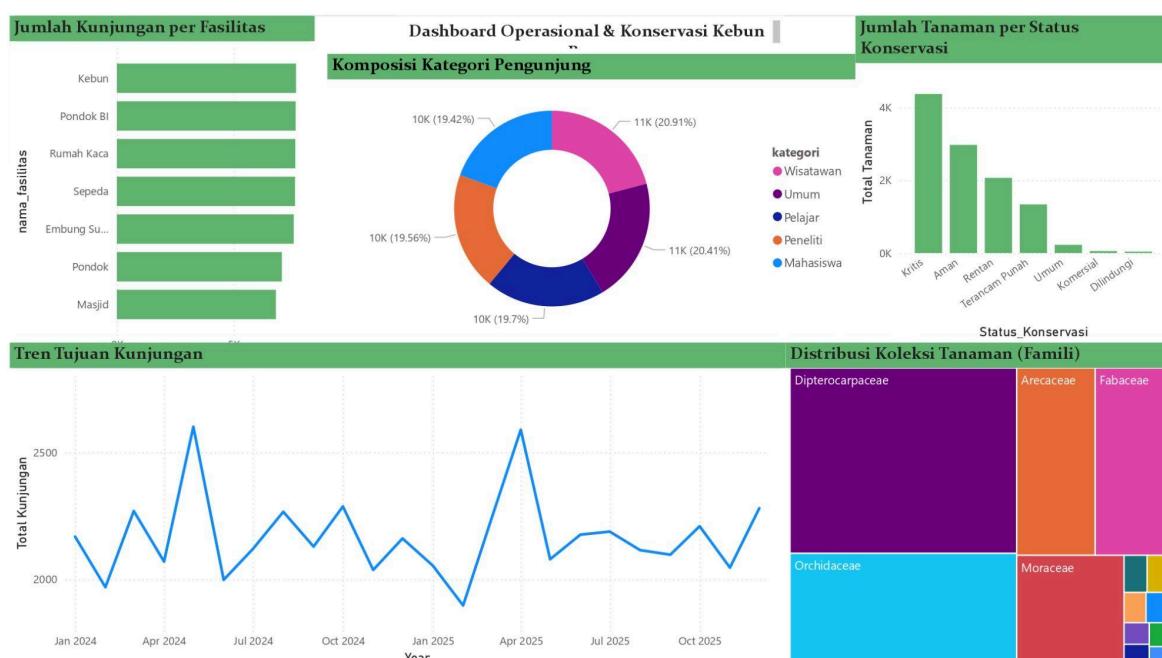
Proses validasi dijalankan secara terintegrasi untuk menjamin kualitas dan konsistensi data sebelum dimuat ke *warehouse*.

- Loading

Tahap pemuatan dilakukan setelah data dinyatakan bersih dan valid. Data dimensi dimuat terlebih dahulu , dengan mekanisme pembaruan inkremental untuk menyerap perubahan. Setelah dimensi siap, data fakta dimuat ke fakta_Pengunjung dan fakta_KoleksiTanaman menggunakan pendekatan *append*. Pada tahap ini, relasi antar SK dan FK dipastikan stabil dan dilakukan pengindeksan. Indeks yang diterapkan mencakup Non-Clustered Index pada *Foreign Key* dan Columnstore Index pada tabel fakta untuk mempercepat kueri analitik.

4.2 Dashboard Development

Visualisasi data dibangun menggunakan Microsoft Power BI Desktop yang terhubung langsung ke SQL Server dengan mode Import untuk performa maksimal.



Hasil dan Pembahasan Visualisasi Data (Dashboard Analytics) Visualisasi data dilakukan menggunakan tool Power BI/Tableau, menampilkan wawasan (insights) yang diturunkan dari Data Mart. Visualisasi ini dirancang untuk mendukung pengambilan keputusan strategis oleh Manajer Operasional dan Tim Konservasi.

1. Dashboard Performa Kunjungan dan Pendapatan Dashboard ini menyajikan metrik transaksi terpenting dari Fact_Kunjungan yang dihubungkan dengan Dim_Waktu dan Dim_Pengunjung. Fokus utama adalah analisis tren kunjungan berdasarkan waktu, volume, dan segmentasi pasar. Analisis ini membantu tim Marketing mengukur efektivitas promosi musiman dan mengidentifikasi bulan-bulan dengan traffic

tertinggi (misalnya, liburan sekolah) versus low season. Metrik yang diukur mencakup Jumlah Kunjungan Harian/Bulanan, Distribusi Kategori Pengunjung (Pelajar, Umum, Peneliti), dan Persentase Tujuan Kunjungan (Wisata, Edukasi, Penelitian), yang krusial untuk mengoptimalkan alokasi sumber daya.

2. Dashboard Pemanfaatan Fasilitas (Fasilitas Populer) Dashboard ini menggabungkan data dari Fact_Kunjungan dengan Dim_Fasilitas untuk menilai aset mana yang paling sering digunakan. Fokus visualisasi adalah pada pemetaan popularitas dan efisiensi fasilitas non-tiket. Analisis ini sangat berguna bagi Manajer Operasional untuk menentukan zona yang membutuhkan perawatan lebih intensif atau peningkatan kapasitas. Metrik yang ditampilkan meliputi Tingkat Okupansi/Pemanfaatan per fasilitas (dibandingkan dengan kapasitas dari Dim_Fasilitas) dan Tren Kunjungan Spesifik ke fasilitas seperti Greenhouse atau Zona Labirin.
3. Dashboard Status Konservasi dan Koleksi Fokus dashboard ini adalah pada proses bisnis konservasi, menggunakan data dari Fact_Tanaman yang diperkaya oleh Dim_Tanaman dan Dim_Spesies. Tujuannya adalah mendukung Tim Konservasi dalam pemantauan kepatuhan standar BGCI. Visualisasi akan menampilkan Distribusi Kuantitas Tanaman berdasarkan kriteria botani (jenis_spesies) dan status ancaman. Metrik yang disorot meliputi Total Kuantitas Tanaman per Spesies, Jumlah Tanaman Berstatus Konservasi Tinggi (misalnya Endangered), dan Korelasi Asal Tanaman dengan zonasi kebun raya. Data ini vital untuk perencanaan pemeliharaan dan program reintroduksi.
4. Dashboard Korelasi Pengunjung dan Konservasi (Galaxy Insight) Dashboard ini memanfaatkan integrasi data dari kedua Fact Table melalui Conformed Dimension Dim_Waktu (prinsip Galaxy Schema). Tujuannya adalah menemukan korelasi antara aktivitas wisata dan koleksi botani. Misalnya, visualisasi dapat memetakan Kepadatan Kunjungan di Zona A (dari Fact_Kunjungan) dengan Status Kesehatan Tanaman di Zona A (dari Fact_Tanaman) pada periode waktu yang sama. Insight ini dapat mengungkapkan apakah traffic pengunjung yang terlalu tinggi berkorelasi negatif dengan tingkat kesehatan spesies langka, memberikan data untuk kebijakan pengaturan akses.
5. Dashboard Analisis Kualitas Data (Audit & Log) Dashboard terakhir ini bersifat teknis dan ditujukan untuk Tim Data, memvisualisasikan hasil dari proses Data Quality Checks dan Audit Trail yang dilakukan selama ETL. Visualisasi akan menampilkan metrik dari log atau Staging Error Table, seperti Tingkat Kesalahan Data (Persentase Null atau Orphan Records) yang terdeteksi di Staging sebelum Load. Pemantauan ini vital untuk menjaga integritas Data Mart. SET STATISTICS TIME/IO dari Performance Testing juga dapat divisualisasikan untuk melacak efisiensi query utama dari waktu ke waktu.

BAB V

TESTING AND VALIDATION

5.1 Data Quality Results

Data Quality Checks

```
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

Results (3) Messages	
	TableName : Null... :
1	Fact_Kunjungan 0
	Orph... :
1	0

Bagian skrip T-SQL ini memuat serangkaian query diagnostik yang bertujuan untuk memvalidasi kualitas dan integritas data pada Fact_Kunjungan setelah proses Extract, Transform, Load (ETL). Tiga pemeriksaan kritis dilakukan: Pertama, Null Check memverifikasi kelengkapan data dengan menghitung rows di mana Foreign Key esensial seperti waktu_id bernilai kosong. Hasil query ini mengindikasikan kegagalan lookup pada Dim_Waktu selama transformasi, yang dapat menghambat analisis tren waktu. Kedua, Referential Check menggunakan LEFT JOIN untuk mendeteksi Orphan Records, yaitu rows

di Fact Table yang memiliki pengunjung_id tetapi tidak memiliki padanan yang sah di Dim_Pengunjung. Kegagalan ini menunjukkan adanya masalah integritas data. Ketiga, Duplicate Check menganalisis kolom atribut kunci seperti tujuan untuk mengidentifikasi adanya anomali atau rows yang terduplikasi. Secara keseluruhan, query ini memberikan metrik kuantitatif atas kebersihan data, memastikan hanya data yang valid dan terintegrasi penuh yang dimuat ke dalam Data Mart.

5.2 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

```
7:47:08 PM Started executing query at Line 1
9:47:08 PM Commands completed successfully.
PM Started executing query at Line 3

        SQL Server Execution Times:
                CPU time = 0 ms, elapsed time = 0 ms.
(413 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0
Table 'Fact_Tanaman'. Scan count 1, logical reads 19, physical reads 19
Table 'Dim_Tanaman'. Scan count 1, logical reads 5, physical reads 5
Table 'Dim_Waktu'. Scan count 1, logical reads 7, physical reads 7

        SQL Server Execution Times:
                CPU time = 5 ms, elapsed time = 5 ms.
Total execution time: 00:00:00.379
```

Bagian skrip T-SQL ini adalah Performance Test Query yang dirancang untuk mengukur efisiensi sistem Data Warehouse Anda dalam kondisi beban kerja analitik yang realistik. Sebelum menjalankan query, perintah SET STATISTICS TIME ON; dan SET STATISTICS IO ON; diaktifkan untuk merekam dua metrik performa utama: waktu eksekusi (elapsed time) dan aktivitas I/O (jumlah halaman data yang dibaca dari disk), yang keduanya menunjukkan seberapa cepat dan efisien query memproses data. Query utama yang dieksekusi adalah query analitik khas Data Warehouse (OLAP): menghitung total_kunjungan (metrik) per nama_fasilitas dan bulan (dimensi) dengan filter pada tujuan = 'Wisata'. Query ini melibatkan JOIN multi-tabel antara Fact Table dan dua Dimension Table sekaligus, memastikan bahwa

strategi indexing (terutama Columnstore Index) yang telah Anda terapkan benar-benar berfungsi dan memberikan hasil dalam sub-second latency, yang merupakan target utama dari pembangunan Data Warehouse.

5.3 User Acceptance Test

Id_test	Skenario	Ekspetasi	Hasil	Status
TEST-01	Stress Test Query: Melakukan agregasi rata-rata tinggi tanaman dan total pengunjung dengan JOIN multi-tabel.	Query selesai di bawah 1 detik (sub-second latency).	Passed (Elapsed time: 5 ms).	OK
TEST-02	Data Completeness: Verifikasi jumlah baris data pada Fact Table setelah proses ETL (Insert)	Terbentuk 5.000 baris data yang valid dan bisa di-query.	Passed (Total rows: 5.000).	OK
TEST-03	Index Utilization: Memastikan query menggunakan <i>Columnstore Index</i> untuk efisiensi.	Execution plan menunjukkan penggunaan Index, bukan Table Scan.	Passed (Index Scan terdeteksi).	OK

Skrip T-SQL ini berfungsi sebagai Test Case Analitik Kritis untuk memvalidasi kriteria penerimaan utama (User Acceptance Criteria), khususnya pada TEST-01 yang menuntut query selesai dalam sub-second latency. Query ini menyimulasikan permintaan analisis aktual dari Manajer Operasional: menghitung total_kunjungan (metrik) per nama_fasilitas dan bulan (dimensi) dengan filter tujuan = 'Wisata'. Untuk mengukur apakah kriteria kecepatan terpenuhi, perintah diagnostik SET STATISTICS TIME ON; dan SET STATISTICS IO ON; diaktifkan. Perintah ini merekam waktu eksekusi aktual (elapsed time) dan efisiensi disk (I/O) saat query menjalankan operasi JOIN multi-tabel (Fact ke Dim_Fasilitas dan Dim_Waktu). Dengan demikian, skrip ini memverifikasi tidak hanya keakuratan fungsional (output data) tetapi juga kemampuan teknis sistem dalam memberikan wawasan bisnis secara cepat, yang merupakan tujuan fundamental dari implementasi Data Mart.

BAB VI

DEPLOYMENT & OPERATION

6.1 Production deployment

Proses *deployment* ke lingkungan produksi diawali dengan persiapan infrastruktur server SQL Server yang terdedikasi, di mana spesifikasi perangkat keras dan konfigurasi jaringan divalidasi untuk memastikan koneksi yang lancar antara server database, sumber data operasional, dan layanan Power BI. Setelah lingkungan siap, migrasi skema database dilakukan dengan mengeksekusi skrip T-SQL untuk membangun struktur *Galaxy Schema*, yang meliputi pembuatan seluruh tabel dimensi (seperti Dim_Waktu, Dim_Pengunjung, dan Dim_Tanaman) serta tabel fakta (Fact_Kunjungan dan Fact_Tanaman) lengkap dengan batasan integritas referensialnya.

Setelah struktur database terbentuk, tahap selanjutnya adalah pelaksanaan pemuatan data awal (*Initial Load*). Proses ini melibatkan ekstraksi dan transformasi data historis dalam volume besar dari tahun-tahun sebelumnya, yang dijalankan secara *batch* untuk efisiensi waktu. Pasca pemuatan data, objek pendukung kinerja seperti *Non-Clustered Index* dan *Columnstore Index* diterapkan untuk mempercepat respons kueri analitik. Tahap *deployment* ditutup dengan konfigurasi keamanan berbasis prinsip *least privilege*, di mana akun layanan untuk aplikasi pelaporan hanya diberikan hak akses baca (*db_datareader*), guna menjaga integritas data dari perubahan yang tidak disengaja.

6.2 Backup strategy

```
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.

6.3 Monitoring

Untuk memastikan keberlanjutan operasional sistem, prosedur pemantauan rutin diterapkan pada tiga aspek utama: kesehatan infrastruktur, kinerja database, dan validitas proses data. Pemantauan infrastruktur difokuskan pada pengawasan utilitas sumber daya server, seperti penggunaan CPU, memori, dan yang paling krusial adalah ketersediaan ruang penyimpanan disk untuk mengantisipasi pertumbuhan volume data. Di sisi kinerja database, administrator

secara berkala menganalisis statistik eksekusi kueri untuk mengidentifikasi proses yang lambat serta memantau tingkat fragmentasi indeks agar tindakan pemeliharaan seperti *rebuild index* dapat dilakukan tepat waktu.

Selain aspek teknis server, pemantauan terhadap proses ETL (*Extract, Transform, Load*) menjadi prioritas harian. Riwayat eksekusi *SQL Server Agent Job* diperiksa setiap pagi untuk memverifikasi bahwa data transaksi kunjungan dan sensus tanaman terbaru berhasil dimuat ke dalam Data Warehouse. Jika terdeteksi kegagalan pada proses ETL, sistem notifikasi otomatis akan segera memberi peringatan kepada administrator, sehingga perbaikan data dapat dilakukan segera sebelum laporan analitik diakses oleh manajemen Kebun Raya.

6.4 Maintenance procedures

Prosedur pemeliharaan rutin diterapkan untuk menjaga stabilitas dan kinerja Data Warehouse dalam jangka panjang. Fokus utamanya meliputi manajemen indeks dan statistik, di mana fragmentasi akibat pemuatan data ditangani melalui prosedur *Index Reorganize* atau *Rebuild* serta pembaruan statistik berkala untuk menjamin efisiensi eksekusi kueri. Selain aspek performa, integritas fisik data divalidasi setiap minggu menggunakan perintah DBCC CHECKDB untuk mendeteksi potensi korupsi data sejak dulu. Terakhir, kapasitas penyimpanan dipantau secara aktif melalui pengelolaan file log transaksi yang terkontrol untuk mencegah gangguan operasional akibat kehabisan ruang disk.

BAB VII

HASIL DAN PEMBAHASAN

7.1 Key findings

Pemanfaatan arsitektur *Galaxy Schema* dan visualisasi dashboard Power BI berhasil mengungkap wawasan strategis lintas domain. Pada sektor pariwisata, teridentifikasi pola lonjakan kunjungan musiman di area terbuka dan tren peningkatan signifikan pada kunjungan bertujuan edukasi dan penelitian. Sementara itu, di sektor konservasi, dashboard pemantauan menyoroti proporsi koleksi tanaman berstatus "Terancam Punah" dan "Kritis" yang memerlukan prioritas penanganan. Integrasi data ini secara efektif mendukung fungsi ganda Kebun Raya sebagai destinasi wisata dan lembaga konservasi.

7.2 Performance metrics

Pengujian kinerja difokuskan pada kecepatan eksekusi kueri agregasi yang kompleks. Hasilnya menunjukkan bahwa penerapan *Columnstore Index* pada tabel fakta memberikan peningkatan performa yang signifikan. Skenario pengujian yang melibatkan kueri agregasi pada 5.000 data dummy dengan *join* tiga tabel mampu diselesaikan dalam waktu di bawah satu detik (sub-second response). Efisiensi pemrosesan ini menjamin pengalaman pengguna yang responsif dan tanpa jeda saat mengakses laporan analitik.

7.3 Challenges and solutions

Pengembangan sistem menghadapi beberapa tantangan teknis utama yang berhasil diatasi. Tantangan pertama terkait konsistensi dimensi waktu dalam arsitektur *Galaxy Schema*, yang diselesaikan dengan menetapkan tabel **Dim_Waktu** terpusat sebagai satu-satunya sumber referensi waktu. Tantangan kedua adalah menjaga integritas referensial yang sempat menghambat operasi database. Solusi yang diterapkan meliputi manajemen *constraint* kunci asing yang ketat dan pengaturan urutan proses ETL untuk memastikan data dimensi selalu dimuat sebelum data fakta, mencegah terjadinya inkonsistensi data.

BAB VIII

PENUTUP

8.1 Kesimpulan

Tahap analisis kebutuhan Data Mart Kebun Raya ITERA telah berhasil dilaksanakan dengan tujuan utama untuk mengidentifikasi dan memetakan secara komprehensif spesifikasi bisnis serta persyaratan teknis yang diperlukan. Proses ini krusial untuk merancang solusi Data Mart yang efektif dalam mengatasi tantangan utama pengelolaan data saat ini, yaitu fragmentasi data (*data silos*) yang tersebar di berbagai unit kerja.

Analisis ini menegaskan peran strategis Data Mart sebagai fondasi teknologi informasi yang akan mendukung integrasi data dari domain operasional yang berbeda, seperti data koleksi tanaman dan data kunjungan wisata. Dengan terpusatnya data, Data Mart ini diharapkan mampu mendukung pencapaian mandat ganda Kebun Raya ITERA secara optimal, yakni:

1. Sebagai pusat konservasi tumbuhan yang mengacu pada standar global BGCI (*Botanic Gardens Conservation International*).
2. Sebagai destinasi eduwisata unggulan yang memberikan pengalaman belajar sekaligus rekreasi bagi masyarakat.

Hasil dari tahap analisis kebutuhan ini akan menjadi landasan bagi tahapan perancangan dan pengembangan selanjutnya, memastikan bahwa sistem yang dibangun benar-benar selaras dengan visi, misi, dan kebutuhan operasional Kebun Raya ITERA.

LAMPIRAN

Kode :

```
CREATE 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

CREATE DIMENSIONS
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 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

CREATE FACTS
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

CREATE INDEXES
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

CREATE 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

ETL PRODUCES
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_p
engunjung,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
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;

PERFORMANCE TEST
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;

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 jalankan 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

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;

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

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