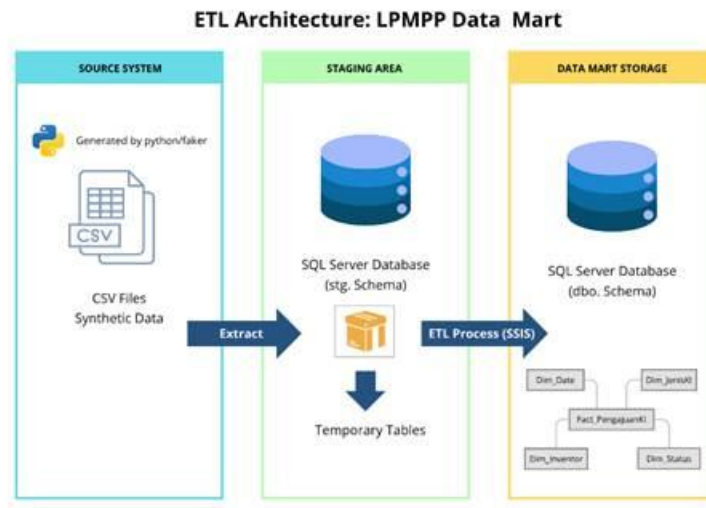


# Complete Documentation Package

## Kelompok 06 LPMPP

### System Architecture:



### 1. Arsitektur ETL

Proses ETL (Extract, Transform, Load) dirancang menggunakan pendekatan **Staging Area** untuk meminimalisir beban pada sistem sumber dan memastikan kualitas data.

- **Extract:** Data diekstrak dari sumber data sintetis (CSV) ke dalam tabel Staging (stg.\*) tanpa perubahan struktur yang signifikan.
- **Transform:** Data di tabel Staging divalidasi tipe datanya (Type Casting), penanganan nilai NULL, dan penyesuaian format.
- **Load:** Data bersih dimuat ke dalam tabel Dimensi dan Fakta (dbo.\*) di Data Warehouse.

### 2. Tools yang Digunakan

- **SSIS (SQL Server Integration Services):** Sebagai *orchestrator* utama alur data.
- **Python Faker:** Sebagai generator data sumber (CSV).
- **SQL Server 2019:** Sebagai database engine untuk Staging dan Production DW.

### 3. Strategi Loading

- **Dimension Load:** Dilakukan terlebih dahulu (Status, JenisKl, Inventor) untuk menjaga referential integrity. Menggunakan metode *Truncate & Load* untuk data master statis.
- **Fact Load:** Dilakukan setelah dimensi terisi. Menggunakan metode *Incremental Insert* (atau *Bulk Insert* untuk initial load).

Data Dictionary:

AutoSave ETL_Mapping_Document • Saved to this PC										Search									
File Home Insert Data Entry Draw Page Layout Formulas Data Review View Developer Help										Comments Share									
Clipboard Font Alignment Number Styles Cells Editing Add-ins																			
G18																			
1	Source File (CSV)	Source Column	Target Table (DW)	Target Column	Transformation Rule / Logic														
2	DIMENSION TABLES																		
3	Dim_JenisKI.csv	JenisKIKey	dbo.Dim_JenisKI	JenisKIKey	Direct Mapping (DT_I4)														
4	Dim_JenisKI.csv	Nama_JenisKI	dbo.Dim_JenisKI	Nama_JenisKI	Direct Mapping (DT_STR)														
5	Dim_Status.csv	StatusKey	dbo.Dim_Status	StatusKey	Direct Mapping (DT_I4)														
6	Dim_Status.csv	Nama_Status	dbo.Dim_Status	Nama_Status	Direct Mapping (DT_STR)														
7	Dim_Inventor.csv	NIP_NIM	dbo.Dim_Inventor	NIP_NIM	Direct Mapping (DT_STR) - Business Key														
8	Dim_Inventor.csv	Nama_Inventor	dbo.Dim_Inventor	Nama_Inventor	Proper Case / Direct Mapping														
9	Dim_Inventor.csv	Nama_Prod	dbo.Dim_Inventor	Nama_Prod	Direct Mapping														
10	Dim_Inventor.csv	Nama_Fakultas	dbo.Dim_Inventor	Nama_Fakultas	Direct Mapping														
11	(System)	-	dbo.Dim_Inventor	StartDate	Derived Column: GETDATE()														
12	(System)	-	dbo.Dim_Inventor	EndDate	Derived Column: NULL														
13	(System)	-	dbo.Dim_Inventor	ICurrent	Derived Column: 1 (Active)														
14	Dim_Date.csv	DateKey	dbo.Dim_Date	DateKey	Direct Mapping (DT_I4)														
15	Dim_Date.csv	FullDate	dbo.Dim_Date	FullDate	Convert to Date (DT_DDATE)														
16	FACT TABLE																		
17	Fact_PengajuanKI.csv	PengajuanKey	dbo.Fact_PengajuanKI	PengajuanKey	Direct Mapping (DT_I4)														
18	Fact_PengajuanKI.csv	DateKey_Daftar	dbo.Fact_PengajuanKI	DateKey_Daftar	LOOKUP to Dim_Date (Fail if Not Found)														
19	Fact_PengajuanKI.csv	DateKey_Granted	dbo.Fact_PengajuanKI	DateKey_Granted	Logic: (LEN(TRIM())=0) ? NULL : Val -> LOOKUP (ignore Failure)														
20	Fact_PengajuanKI.csv	DateKey_Kadaluwarsa	dbo.Fact_PengajuanKI	DateKey_Kadaluwarsa	Logic: (LEN(TRIM())=0) ? NULL : Val -> LOOKUP (ignore Failure)														
21	Fact_PengajuanKI.csv	JenisKIKey	dbo.Fact_PengajuanKI	JenisKIKey	LOOKUP to Dim_JenisKI (Fail if Not Found)														
22	Fact_PengajuanKI.csv	StatusKey	dbo.Fact_PengajuanKI	StatusKey	LOOKUP to Dim_Status (Fail if Not Found)														
23	Fact_PengajuanKI.csv	BiayaPendaftaran	dbo.Fact_PengajuanKI	BiayaPendaftaran	Logic: (LEN(TRIM())=0) ? NULL : Val -> Convert to DECIMAL (DT_CY)														
24	Fact_PengajuanKI.csv	JumlahPengajuan	dbo.Fact_PengajuanKI	JumlahPengajuan	Logic: (LEN(TRIM())=0) ? 0 : Val -> Convert to INT (DT_I4)														
25	Fact_PengajuanKI.csv	JumlahPaten	dbo.Fact_PengajuanKI	JumlahPaten	Logic: (LEN(TRIM())=0) ? 0 : Val -> Convert to INT (DT_I4)														
26	(System)	-	dbo.Fact_PengajuanKI	LoadDate	SQL Default: GETDATE()														
27	BRIDGE TABLE																		
28	Bridge.csv	PengajuanKey	dbo.Bridge_Pengajuan_Inventor	PengajuanKey	Validated via LOOKUP to Fact_PengajuanKI														
29	Bridge.csv	InventorKey (NIP)	dbo.Bridge_Pengajuan_Inventor	InventorKey	LOOKUP to Dim_Inventor (Match NIP, Return InventorKey)														
30	Bridge.csv	Peran_Inventor	dbo.Bridge_Pengajuan_Inventor	Peran_Inventor	Direct Mapping														
31																			

Source File (CSV)	Source Column	Target Table (DW)	Target Column	Transformation Rule / Logic
DIMENSION TABLES				
Dim_JenisKI.csv	JenisKIKey	dbo.Dim_JenisKI	JenisKIKey	Direct Mapping (DT_I4)
Dim_JenisKI.csv	Nama_JenisKI	dbo.Dim_JenisKI	Nama_JenisKI	Direct Mapping (DT_STR)
Dim_Status.csv	StatusKey	dbo.Dim_Status	StatusKey	Direct Mapping (DT_I4)

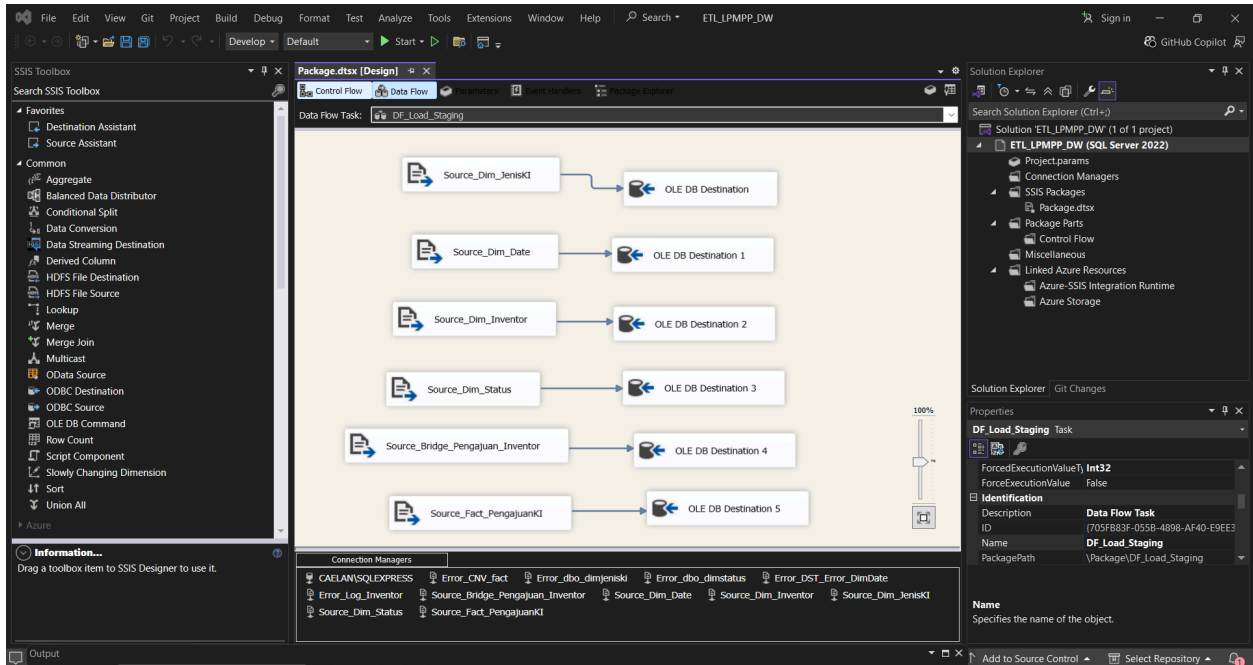
Dim_Status .csv	Nama_Status	dbo.Dim_Status	Nama_Stat us	Direct Mapping (DT_STR)
Dim_Invent or.csv	NIP_NIM	dbo.Dim_Invent or	NIP_NIM	Direct Mapping (DT_STR) - Business Key
Dim_Invent or.csv	Nama_Invent or	dbo.Dim_Invent or	Nama_Inve ntor	Proper Case / Direct Mapping
Dim_Invent or.csv	Nama_Prodi	dbo.Dim_Invent or	Nama_Pro di	Direct Mapping
Dim_Invent or.csv	Nama_Fakult as	dbo.Dim_Invent or	Nama_Fak ultas	Direct Mapping
(System)	-	dbo.Dim_Invent or	StartDate	Derived Column: GETDATE()
(System)	-	dbo.Dim_Invent or	EndDate	Derived Column: NULL
(System)	-	dbo.Dim_Invent or	IsCurrent	Derived Column: 1 (Active)
Dim_Date.c sv	DateKey	dbo.Dim_Date	DateKey	Direct Mapping (DT_I4)
Dim_Date.c sv	FullDate	dbo.Dim_Date	FullDate	Convert to Date (DT_DBDATE)

FACT TABLE				
Fact_Peng ajuanKI.csv	PengajuanKey	dbo.Fact_Peng ajuanKI	Pengajuan Key	Direct Mapping (DT_I4)
Fact_Peng ajuanKI.csv	DateKey_Daftar	dbo.Fact_Peng ajuanKI	DateKey_D aftar	LOOKUP to Dim_Date (Fail if Not Found)
Fact_Peng ajuanKI.csv	DateKey_Granted	dbo.Fact_Peng ajuanKI	DateKey_G ranted	Logic: (LEN(TRIM)=0) ? NULL : Val -> LOOKUP (Ignore Failure)
Fact_Peng ajuanKI.csv	DateKey_Kadaluwarsa	dbo.Fact_Peng ajuanKI	DateKey_K adaluwarsa	Logic: (LEN(TRIM)=0) ? NULL : Val -> LOOKUP (Ignore Failure)
Fact_Peng ajuanKI.csv	JenisKIKey	dbo.Fact_Peng ajuanKI	JenisKIKey	LOOKUP to Dim_JenisKI (Fail if Not Found)
Fact_Peng ajuanKI.csv	StatusKey	dbo.Fact_Peng ajuanKI	StatusKey	LOOKUP to Dim_Status (Fail if Not Found)
Fact_Peng ajuanKI.csv	BiayaPendaftaran	dbo.Fact_Peng ajuanKI	BiayaPend aftaran	Logic: (LEN(TRIM)=0) ? NULL : Val -> Convert to DECIMAL (DT_CY)
Fact_Peng ajuanKI.csv	JumlahPengajuan	dbo.Fact_Peng ajuanKI	JumlahPen gajuan	Logic: (LEN(TRIM)=0) ? 0 : Val -> Convert to INT (DT_I4)

Fact_PengajuanKI.csv	JumlahPaten	dbo.Fact_PengajuanKI	JumlahPaten	Logic: (LEN(TRIM)==0) ? 0 : Val -> Convert to INT (DT_I4)
(System)	-	dbo.Fact_PengajuanKI	LoadDate	SQL Default: GETDATE()
BRIDGE TABLE				
Bridge.csv	PengajuanKey	dbo.Bridge_Pengajuan_Inventor	PengajuanKey	Validated via LOOKUP to Fact_PengajuanKI
Bridge.csv	InventorKey (NIP)	dbo.Bridge_Pengajuan_Inventor	InventorKey	LOOKUP to Dim_Inventor (Match NIP, Return InventorKey)
Bridge.csv	Peran_Inventor	dbo.Bridge_Pengajuan_Inventor	Peran_Inventor	Direct Mapping

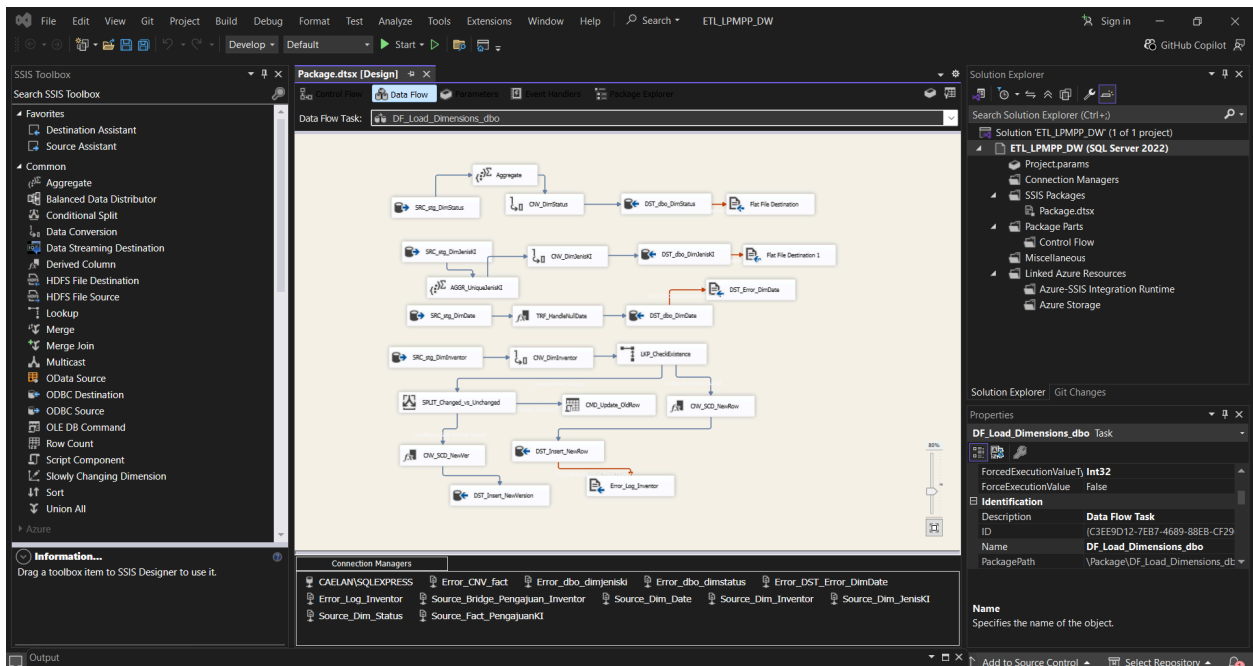
## ETL Documentation:

Tahap load staging:



Sesuai poin 'Extract dari Source' pada modul, kami melakukan ekstraksi data dari Flat File (CSV) menuju tabel Staging (stg) di database. Ini dilakukan untuk menampung data mentah sebelum diproses lebih lanjut.

## Tahap Load Dimensions:



Data Flow Task (DFT) 2 bertanggung jawab untuk memuat dan memproses data dimensi dari Staging Area ke tabel Dimensi di Data Warehouse (DM\_LPMPP\_DW). Dimensi yang dimuat meliputi Dim\_JenisKI, Dim\_Status, Dim\_Date, dan Dim\_Inventor.

Catatan Khusus: Implementasi SCD Tipe 2 pada Dim\_Inventor

Aspek	Definisi Awal (Data Dictionary)	Implementasi Aktual (DFT 2)
Model	Slowly Changing Dimension (SCD) Tipe 1 / Tipe 0	SCD Tipe 2 (Historical Tracking)
Kolom DDL	InventorKey, NIP_NIM, Nama_Inventor, Nama_Prodi, Nama_Fakultas	Ditambah: StartDate, EndDate, IsCurrent
Justifikasi	Perubahan model dilakukan untuk memenuhi kebutuhan bisnis pelaporan historis. Penting untuk mengetahui status (Nama_Prodi, Nama_Fakultas) seorang inventor pada saat pengajuan Kekayaan Intelektual (KI) di masa lalu, yang hanya bisa dicapai dengan SCD Tipe 2. Dokumen DDL dan Data Dictionary wajib diperbarui.	

Detail Alur Logika Dim\_Inventor (SCD Tipe 2)

Alur ini memisahkan data inventor menjadi tiga kategori: Baris Baru (Insert), Baris Berubah (Update + Insert New Version), dan Baris Tidak Berubah (Abaikan).

Tahap	Komponen SSIS	Output	Aksi/Logika

1. Pencarian	Lookup (LKP_CheckExistence)	Match & No Match	Mencari NIP_NIM inventor dari <i>staging</i> di tabel dbo.Dim_Inventor (DW).
2. Insert Baris Baru	No Match Output	→ CNV_SCD_NewRow	Jalur 1 (Baris Baru): Data inventor belum ada di DW.
3. Konversi (Baru)	Derived Column (CNV_SCD_NewRow)	→ DST_Insert_NewRow	Membuat 3 kolom SCD wajib (StartDate=GETDATE(), EndDate=NULL, IsCurrent=1).
4. Insert Tujuan (Baru)	OLE DB Destination (DST_Insert_NewRow)	-	INSERT data inventor baru (dengan kolom SCD lengkap) ke dbo.Dim_Inventor.
5. Pemeriksaan Perubahan	Match Output → Conditional Split (SPLIT_Changed_vs_Unchanged)	Output Changed & Output Unchanged	Jalur 2 (Baris Lama): Membandingkan Nama_Prodi dari <i>staging</i> dengan Old_Nama_Prodi dari Lookup.



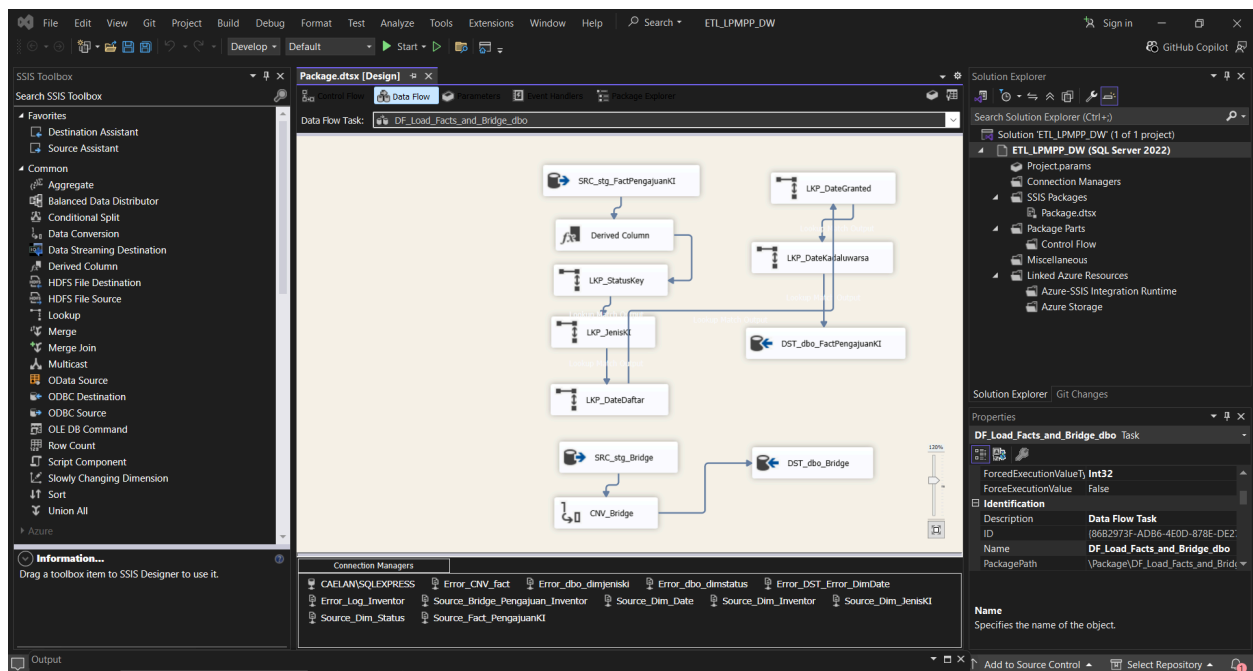
6. Update Baris Lama	Output Changed → OLE DB Command (CMD_Update_OldRow)	-	Menggunakan SK_Exist untuk menjalankan perintah SQL UPDATE pada baris lama: SET IsCurrent = 0, EndDate = GETDATE() WHERE InventorKey = ?. (Menutup riwayat lama).
7. Insert Versi Baru	Output Changed → Derived Column (CNV_SCD_NewVersion)	→ DST_Insert_NewVersion	Jalur 3 (Baris Berubah): Data yang sama (yang baru di-update di Tahap 6) disalurkan ke Derived Column untuk membuat versi baru (StartDate=GETDATE(), EndDate=NULL, IsCurrent=1).
8. Insert Tujuan (Versi Baru)	OLE DB Destination (DST_Insert_NewVersion)	-	INSERT versi data terbaru (dengan kolom SCD lengkap) ke dbo.Dim_Inventor.
9. Baris Tidak Berubah	Output Unchanged	Abaikan	Data lama dan data baru identik. Baris ini dibuang dari

			aliran data (tidak ada aksi).
--	--	--	-------------------------------

Dengan alur ini, setiap perubahan pada inventur akan mencatat riwayat versi lama dan memasukkan versi baru sebagai baris yang aktif (IsCurrent=1).

### Tahap Load Facts and Bridge:

DFT 3 adalah tahap terakhir dalam proses ETL, bertugas memproses data transaksi (Fact\_PengajuanKI) dan tabel penghubung (Bridge\_Pengajuan\_Inventor) dari Staging Area (stg) menuju tabel tujuan (dbo). Tugas utama adalah validasi Foreign Key menggunakan dimensi yang sudah terisi di DFT 2.



#### I. Pipeline 1: Fact\_PengajuanKI (Data Transaksi)

1. Ekstraksi dan Konversi (SRC\_stg\_FactPengajuanKI → CNV\_FactPengajuanKI):
  - Aksi: Mengkonversi semua *keys* (DateKey, StatusKey, JenisKIKey) dan *measures* (Jumlah..., BiayaPendaftaran) dari VARCHAR ke DT\_I4 (INT) dan DT\_NUMERIC (DECIMAL 18, 2).
2. Validasi Foreign Key (Chained Lookups):
  - Aksi: Data hasil konversi melewati rantai Lookup Transformation yang diatur ke "Fail component" jika tidak ada kecocokan. Ini memastikan Integritas Referensial.
  - Rantai Lookup:

- LKP\_StatusKey → Validasi keberadaan StatusKey di dbo.Dim\_Status.
- LKP\_JenisKI → Validasi keberadaan JenisKIDeKey di dbo.Dim\_JenisKI.
- LKP\_DateDaftar / LKP\_DateGranted / LKP\_DateKadaluwarsa → Validasi keberadaan kunci tanggal di dbo.Dim\_Date. (Diatur untuk mengizinkan NULL pada DateKey\_Granted/Kadaluwarsa).

3. Load Final (Destination):

- Komponen: OLE DB Destination (DST\_dbo\_FactPengajuanKI).
- Pengaturan Kritis: FastLoadTableLock = True dan FastLoadCheckConstraints = False untuk kecepatan *load* maksimal.
- Mapping: Semua kolom Conv\_... dipetakan ke kolom tujuan (dbo.Fact\_PengajuanKI). Kolom LoadDate dibiarkan kosong agar diisi oleh DEFAULT GETDATE() SQL Server.

## II. Pipeline 2: Bridge\_Pengajuan\_Inventor (Tabel Penghubung)

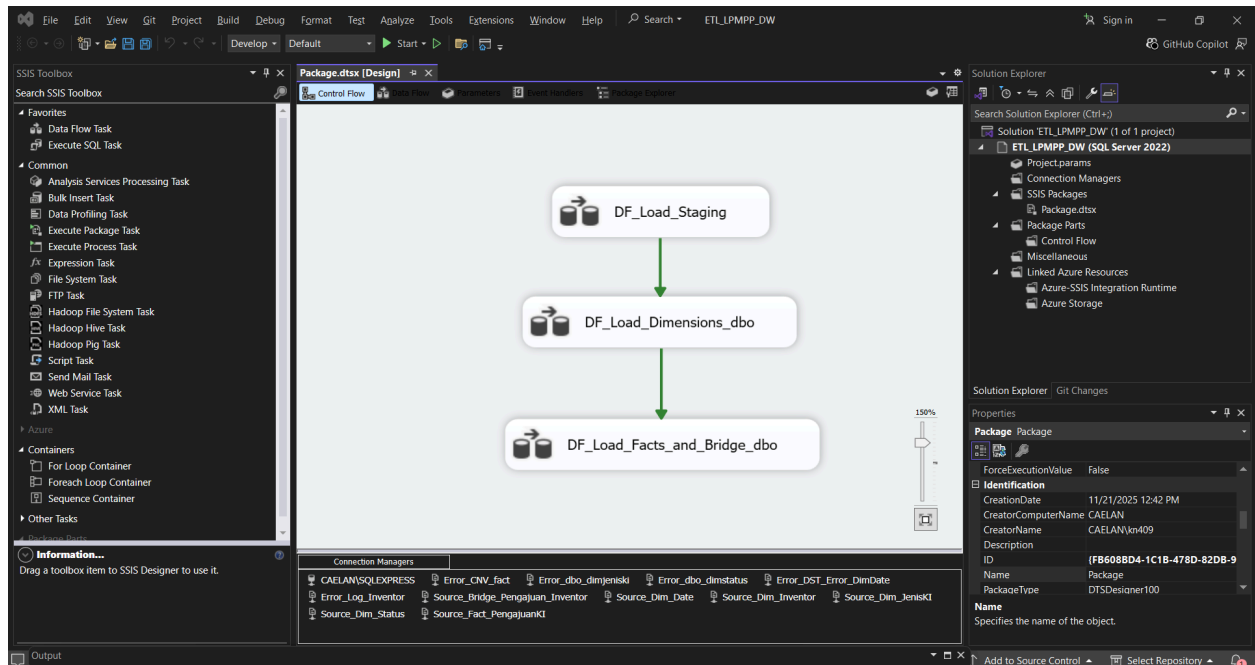
1. Ekstraksi dan Konversi (SRC\_stg\_Bridge → CNV\_Bridge):

- Aksi: Mengkonversi PengajuanKey dan InventorKey menjadi DT\_I4 (INT) dan Peran\_Inventor menjadi DT\_STR (VARCHAR 100).
- Catatan: Pipeline ini adalah *data flow* yang mandiri (tidak menerima input panah dari Fact) dan dimulai dari *OLE DB Source* stg.Bridge\_Pengajuan\_Inventor.

2. Load Final (Destination):

- Komponen: OLE DB Destination (DST\_dbo\_Bridge).
- Aksi: Memuat data langsung ke dbo.Bridge\_Pengajuan\_Inventor. Karena tabel ini berfungsi sebagai gabungan kunci, ia tidak memerlukan *Lookup* di sini.
- Pengaturan Kritis: FastLoadTableLock = True.

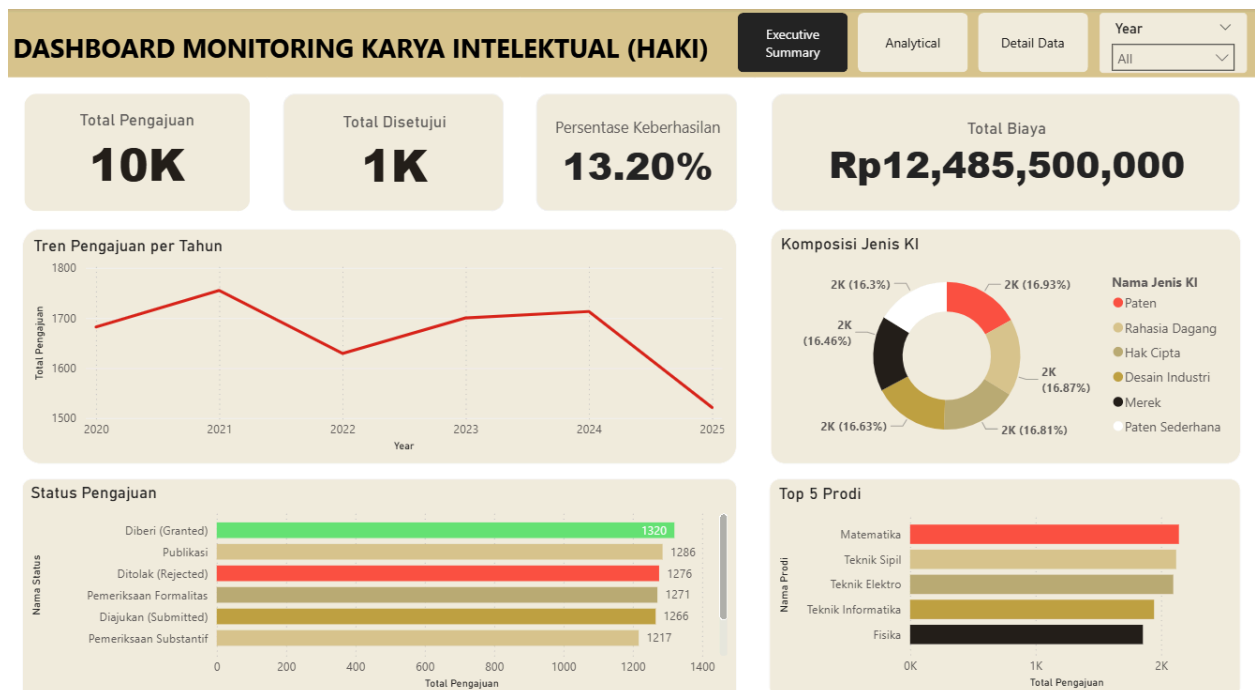
Alur Data Flow Keseluruhan:



Alur kerja ETL diatur menggunakan Control Flow yang terdiri dari tiga tahapan utama secara berurutan: (1) Execute SQL Task untuk membersihkan (truncate) tabel staging agar tidak terjadi duplikasi data, (2) Data Flow Task untuk memuat data dari CSV ke Staging, dan (3) Data Flow Task untuk memuat data dari Staging ke Data Warehouse (Fact & Dimension)

## User Manual:

### Dashboard 1 Executive View



Tampilan Executive Summary untuk pemantauan KPI strategis.

Tujuan: Menyajikan gambaran kinerja secara makro dalam waktu singkat ( $\pm 5$  detik) bagi pimpinan (Rektor/Kepala LPMPP).

## **A. KPI Cards (Indikator Kinerja Utama)**

### **1. Total Pengajuan**

Definisi: Jumlah keseluruhan transaksi pendaftaran HAKI yang tercatat dalam sistem.

Fungsi: Menggambarkan beban kerja administrasi serta volume inovasi yang dihasilkan civitas akademika.

Sumber Data: *COUNT(Fact\_PengajuanKI)*.

### **2. Total Disetujui (Granted)**

Definisi: Jumlah pengajuan dengan status akhir “Diberi/Granted”.

Fungsi: Menunjukkan tingkat keberhasilan output nyata dan menjadi indikator yang relevan untuk pelaporan mutu (misal ke BAN-PT).

Sumber Data: *CALCULATE(COUNT, Status = 'Diberi')*.

### **3. Tingkat Keberhasilan (Success Rate)**

Definisi: Persentase pengajuan yang berhasil disetujui dibandingkan total pengajuan.

Fungsi: Mengukur kualitas dokumen/usulan yang diajukan. Nilai rendah mengindikasikan banyak berkas yang tidak memenuhi standar atau ditolak.

Rumus:  $(Total\ Granted \div Total\ Pengajuan) \times 100\%$ .

### **4. Total Pendapatan (Biaya/PNBP)**

Definisi: Total nilai biaya pendaftaran yang disetor dalam bentuk PNBP.

Fungsi: Mengukur kontribusi finansial unit HAKI terhadap PNBP institusi.

## **B. Komponen Visual**

### **1. Grafik Tren Tahunan (Line Chart)**

Visual: Grafik garis yang menampilkan tren jumlah pengajuan tahun 2020–2024.

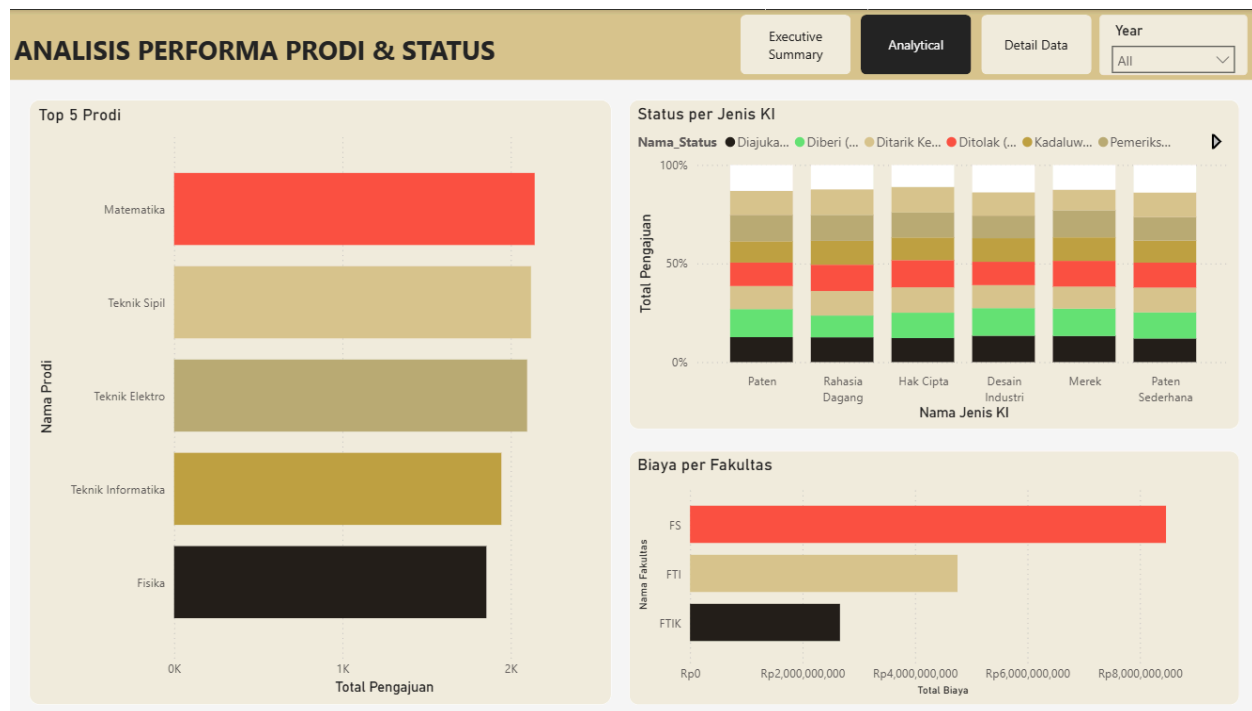
Fungsi: Mengidentifikasi perkembangan inovasi kampus, apakah mengalami peningkatan, stagnasi, atau penurunan.

### **2. Komposisi Jenis KI (Donut Chart)**

Visual: Diagram cincin yang memperlihatkan proporsi jenis KI (Paten, Paten Sederhana, Hak Cipta).

Fungsi: Menunjukkan fokus dominan inovasi, apakah lebih condong pada karya tulisan atau inovasi berbasis teknologi.

## Dashboard 2 Analytical View



Tampilan Analytical View untuk perbandingan performa antar unit kerja.

Tujuan: Memberikan pandangan komparatif untuk kebutuhan evaluasi oleh Manajer/Kepala Pusat.

### 1. Top 5 Program Studi Paling Produktif (Horizontal Bar Chart)

Visual: Lima batang horizontal dengan nilai tertinggi mewakili program studi yang paling banyak mengajukan HAKI.

Fungsi: Mengidentifikasi unit paling produktif (dapat diberikan apresiasi) serta unit yang kurang aktif (memerlukan sosialisasi tambahan).

### 2. Status Pengajuan per Jenis KI (Stacked Column Chart)

Visual: Grafik batang per jenis KI dengan segmentasi warna untuk setiap status (misal: Granted, Rejected, Proses).

Fungsi: Menilai kualitas pengajuan tiap jenis KI. Contoh: Jika “Paten Sederhana” memiliki penolakan tinggi, dapat menjadi dasar evaluasi prosedur maupun pendampingan.

### 3. Kontribusi Biaya per Fakultas (Bar Chart)

Visual: Perbandingan total biaya pendaftaran HAKI antar fakultas.

Fungsi: Menggambarkan distribusi investasi penelitian. Fakultas dengan kontribusi besar menunjukkan intensitas dan komitmen riset yang lebih tinggi.

## Dashboard 3 Detail View

DETAIL RIWAYAT PENGAJUAN								Executive Summary	Analytical	Detail Data	Year All
Nama_Inventor <input type="text"/> Search <input type="checkbox"/> Ade Wahyudin <input type="checkbox"/> Adhijarja Suryono											
Year	Quarter	Month	Day	Nama_Inventor	Nama_Prodi	Nama_JenisKI	Nama_Status	Sum of BiayaPendaftaran			
2020	Qtr 1	January	1	Daliono Lazuardi	Teknik Elektro	Merek	Ditolak (Rejected)	1,000,000.00			
2020	Qtr 1	January	1	dr. Elisa Budiyoanto, M.Tl.	Sains Data	Hak Cipta	Kadaluwarsa (Expired)	1,500,000.00			
2020	Qtr 1	January	1	dr. Elisa Budiyoanto, M.Tl.	Sains Data	Merek	Publikasi	2,000,000.00			
2020	Qtr 1	January	1	Dr. Saiful Hariyah	Teknik Informatika	Hak Cipta	Kadaluwarsa (Expired)	1,500,000.00			
2020	Qtr 1	January	1	dr. Zizi Prabowo, S.Pd	Matematika	Hak Cipta	Kadaluwarsa (Expired)	1,500,000.00			
2020	Qtr 1	January	1	dr. Zizi Prabowo, S.Pd	Matematika	Paten Sederhana	Ditolak (Rejected)	1,500,000.00			
2020	Qtr 1	January	1	Hani Saputra	Matematika	Paten Sederhana	Ditolak (Rejected)	1,500,000.00			
2020	Qtr 1	January	1	Kamila Zulkarnain	Teknik Elektro	Paten Sederhana	Diajukan (Submitted)	1,000,000.00			
2020	Qtr 1	January	1	Taufan Namaga	Fisika	Paten Sederhana	Ditolak (Rejected)	1,500,000.00			
2020	Qtr 1	January	1	Tgk. Tiara Nuraini	Teknik Elektro	Merek	Ditolak (Rejected)	1,000,000.00			
2020	Qtr 1	January	1	Umar Pratiwi	Matematika	Merek	Ditolak (Rejected)	1,000,000.00			
2020	Qtr 1	January	1	Wulan Salahudin, S.Ked	Teknik Sipil	Paten Sederhana	Diajukan (Submitted)	1,000,000.00			
2020	Qtr 1	January	2	Cut Widya Nasyidah	Biologi	Hak Cipta	Diajukan (Submitted)	1,000,000.00			
2020	Qtr 1	January	2	Drs. Sabar Sihombing	Biologi	Merek	Pemeriksaan Formalitas	1,000,000.00			
2020	Qtr 1	January	2	Drs. Umi Mayasari, S.Pd	Fisika	Merek	Ditolak (Rejected)	1,000,000.00			
2020	Qtr 1	January	2	Hasta Sinaga	Teknik Sipil	Rahasia Dagang	Ditolak (Rejected)	1,500,000.00			
2020	Qtr 1	January	2	Hj. Syahrini Salahudin	Matematika	Merek	Kadaluwarsa (Expired)	2,000,000.00			
2020	Qtr 1	January	2	Kasih Puspasari	Teknik Sipil	Hak Cipta	Diberi (Granted)	1,500,000.00			
Total								12,485,500,000.00			

Tampilan Detail Operational untuk penelusuran riwayat pengajuan.

Tujuan: Mendukung kebutuhan operasional staf admin dalam pencarian dan verifikasi data secara terperinci.

## 1. Tabel Riwayat Pengajuan (Table)

Kolom Informasi:

- Tanggal Daftar: Waktu pengajuan diterima.
- Inventor: Nama dosen pengaju.
- Judul/Jenis KI: Objek KI yang didaftarkan.
- Status Terakhir: Posisi proses terkini (misal: Substantif, Formalitas, atau Selesai).

Fungsi: Menjadi pengganti laporan Excel manual dan menyediakan akses langsung terhadap data mentah.

## 2. Pencarian Nama Dosen (Search Slicer)

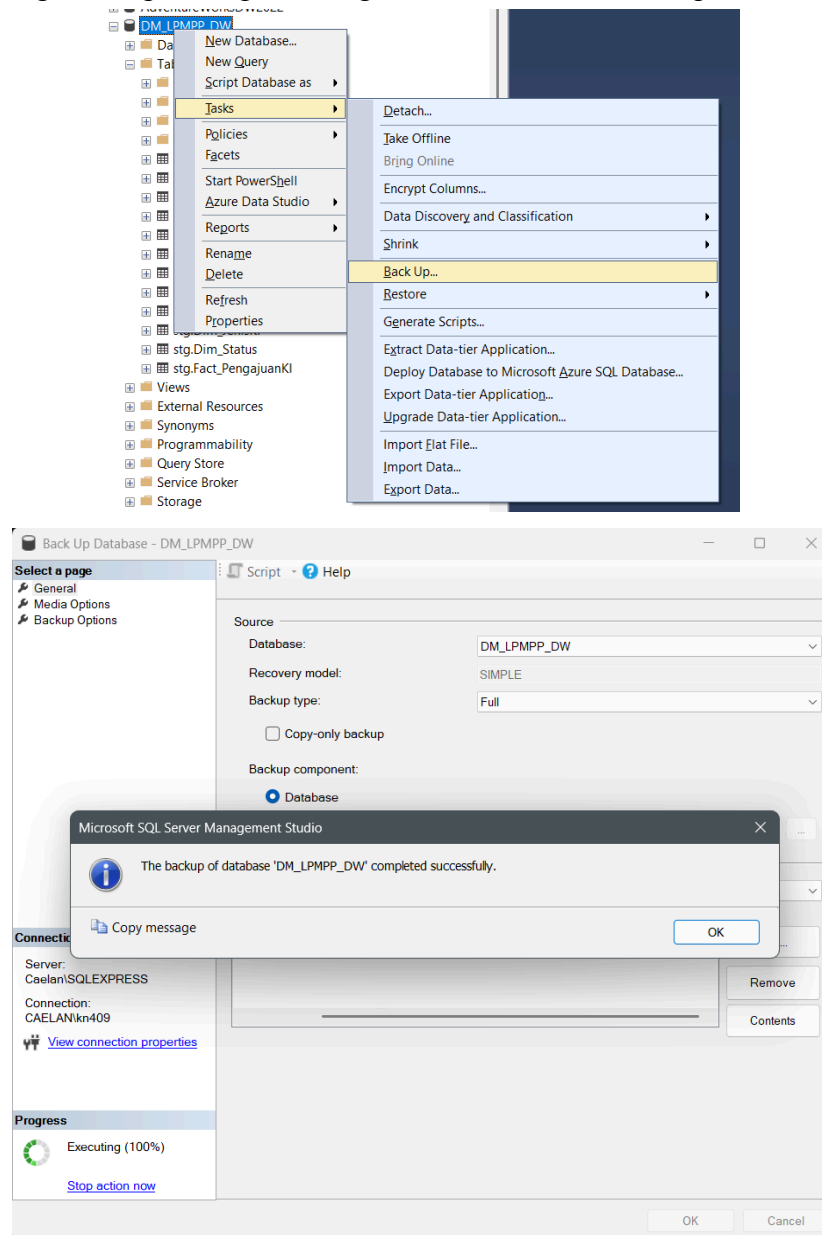
Visual: Kotak pencarian yang dapat memfilter tabel berdasarkan nama dosen.

Fungsi: Mendukung proses audit dan pelayanan. Ketika ada permintaan informasi status oleh dosen, admin dapat melakukan pencarian cepat untuk menampilkan seluruh riwayat pengajuannya.

## Operations Manual:

## 1 Backup Strategy

Untuk menjamin ketersediaan data (Data Availability) dan meminimalisir risiko kehilangan data (Data Loss), strategi backup diterapkan dengan kombinasi Full Backup dan Differential Backup.



Proses Backup Database Full dilakukan sebagai strategi migrasi data dan Disaster Recovery Plan. File .bak disimpan pada lokasi penyimpanan aman.

Berikut adalah jadwal eksekusi backup otomatis yang telah dikonfigurasi:

Tabel 4.1 Jadwal Backup Database



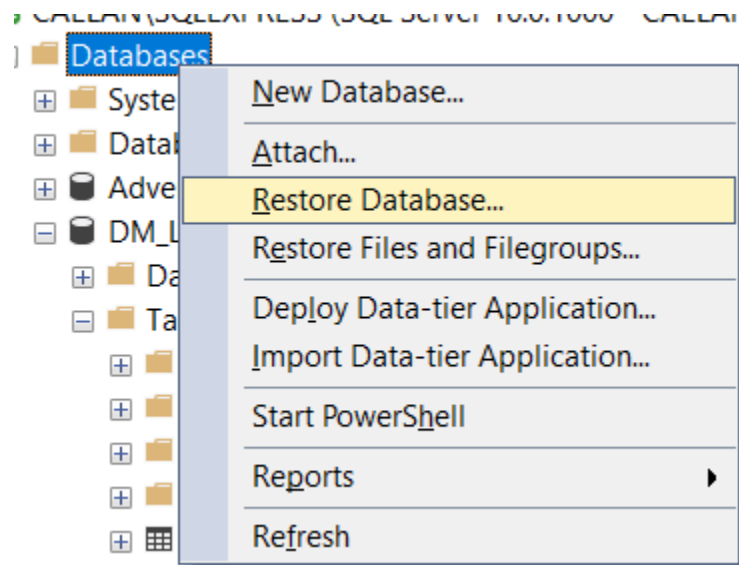
No	Tipe Backup	Frekuensi	Waktu Eksekusi	Retensi (Masa Simpan)	Tujuan & Keterangan
1	<b>Full Database Backup</b>	Mingguan	Minggu, 00:00 WIB	1 Bulan	Menyalin seluruh data dan objek database secara lengkap. Dilakukan saat <i>off-peak hours</i> untuk menghindari beban server.
2	<b>Differential Backup</b>	Harian	Senin - Sabtu, 00:00 WIB	1 Minggu	Hanya menyalin data yang berubah sejak Full Backup terakhir. Ukuran file lebih kecil dan proses lebih cepat.
3	<b>Transaction Log Backup</b>	Per 4 Jam	Setiap 4 jam (04:00, 08:00, dst)	2 Hari	Menyalin log transaksi untuk memungkinkan <i>Point-in-Time Recovery</i> jika terjadi kerusakan mendadak.

## 2 Recovery Procedure

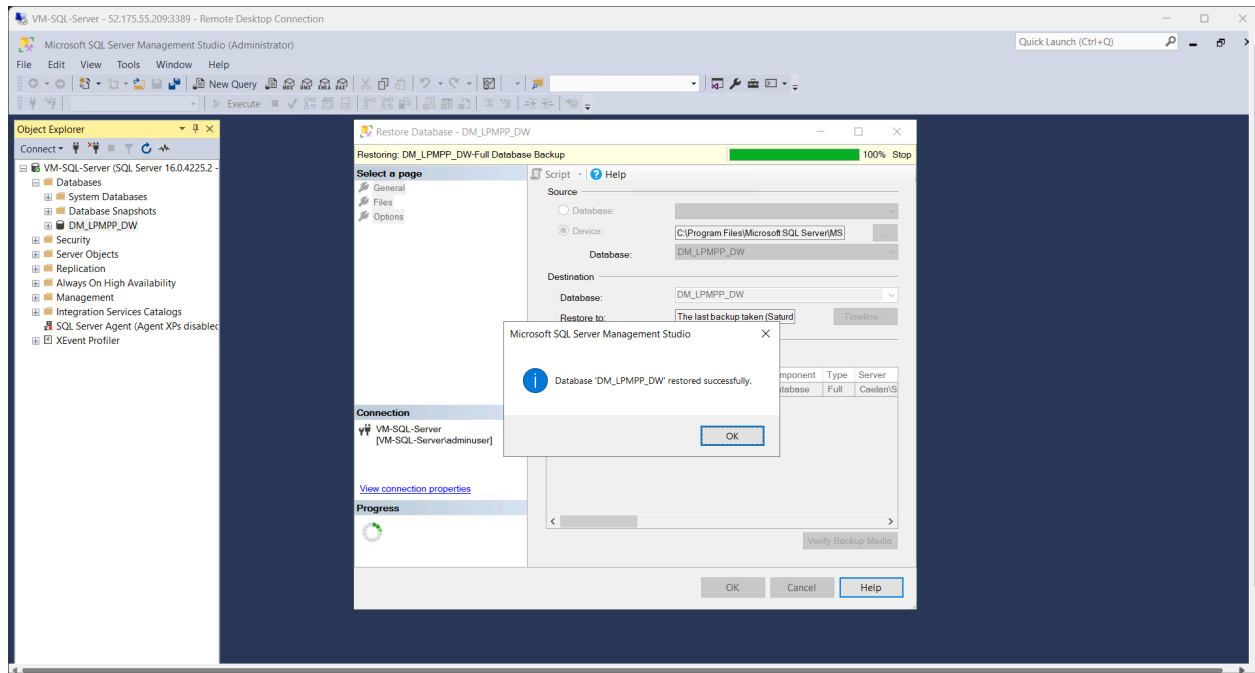
Prosedur ini digunakan untuk mengembalikan kondisi database ke titik terakhir yang stabil apabila terjadi kegagalan sistem (System Failure) atau korupsi data.

### Langkah-langkah Pemulihan (Restore):

1. **Persiapan:** Pastikan tidak ada koneksi aktif ke database target (DM\_LPMPP\_DW). Jika perlu, ubah database ke mode *Single User*.
2. **Akses SSMS:** Buka SQL Server Management Studio (SSMS) dan login sebagai Admin (User\_Admin).
3. **Inisiasi Restore:**
  - Klik kanan pada folder **Databases**.
  - Pilih **Restore Database...**



4. **Pilih Sumber Cadangan:**
  - Pada opsi *Source*, pilih **Device**.
  - Klik tombol (...) dan arahkan ke lokasi file backup (.bak) yang valid
5. **Konfigurasi Opsi:**
  - Pada tab *Options*, centang **Overwrite the existing database (WITH REPLACE)**.
  - Pastikan opsi *Close existing connections to destination database* dicentang.
6. **Eksekusi:** Klik **OK** dan tunggu hingga proses selesai.
7. **Verifikasi:** Setelah muncul pesan "*Database restored successfully*", lakukan pengecekan data pada tabel Fakta untuk memastikan integritas data.



Gambar restore database yang di lakukan di cloud database

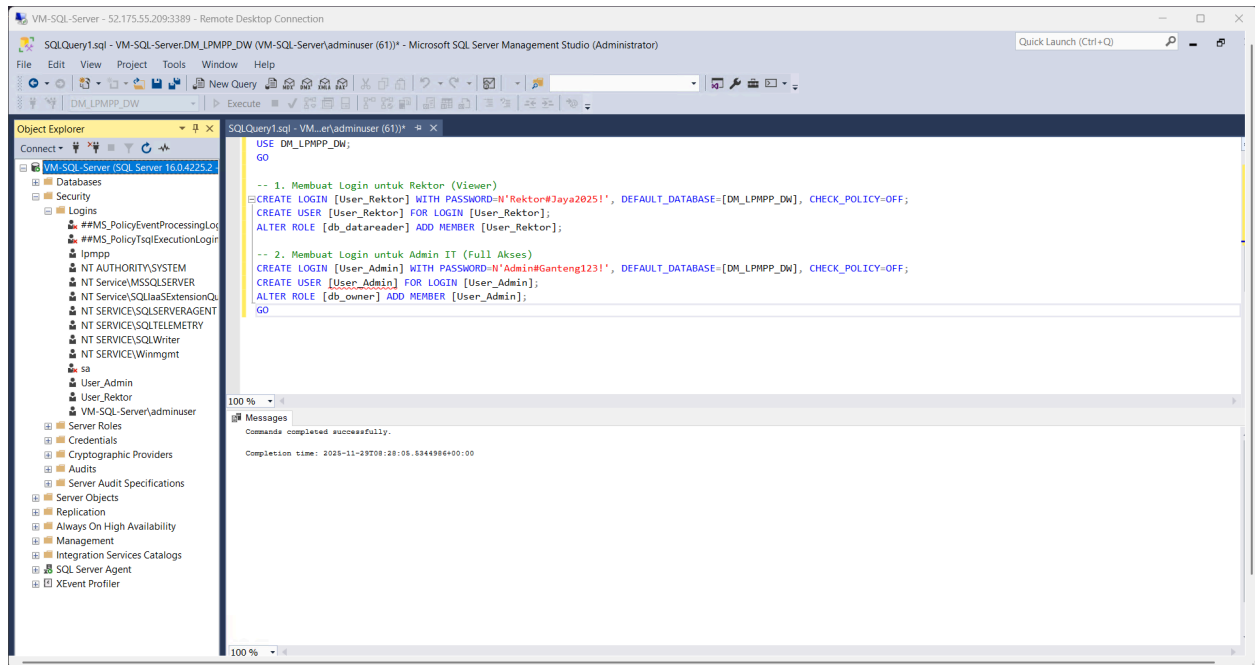
## Security Documentation:

### User Access Matrix

Role	User	Hak Akses	Keterangan
Executive	Rektor, Ka. LPMPP	db_datareader	Hanya bisa melihat dashboard (Read-Only).
Developer	Tim IT Data	db_owner	Bisa mengubah tabel dan menjalankan ETL.

## 2 Implementation Evidence

Implementasi dilakukan menggunakan SQL Server Role-Based Security.



### 3 Code

```
USE DM_LPMPP_DW;

GO

-- 1. Membuat Login untuk Rektor (Viewer)
CREATE LOGIN [User_Rektor] WITH PASSWORD=N'YourStrongPasswordHere',
DEFAULT_DATABASE=[DM_LPMPP_DW];
CREATE USER [User_Rektor] FOR LOGIN [User_Rektor];
-- Kasih hak akses HANYA BISA BACA (Reader)
ALTER ROLE [db_datareader] ADD MEMBER [User_Rektor];

-- 2. Membuat Login untuk Admin IT (Full Akses)
CREATE LOGIN [User_Admin] WITH PASSWORD=N'AdminStrong!',
DEFAULT_DATABASE=[DM_LPMPP_DW];
CREATE USER [User_Admin] FOR LOGIN [User_Admin];
-- Memberi hak akses FULL (Owner)
ALTER ROLE [db_owner] ADD MEMBER [User_Admin];
GO
```