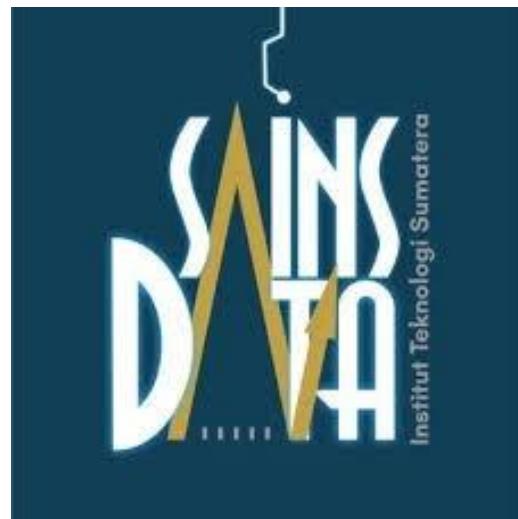


LAPORAN TUGAS BESAR

PERGUDANGAN DATA

MISI 3



Oleh :

Kelompok 10

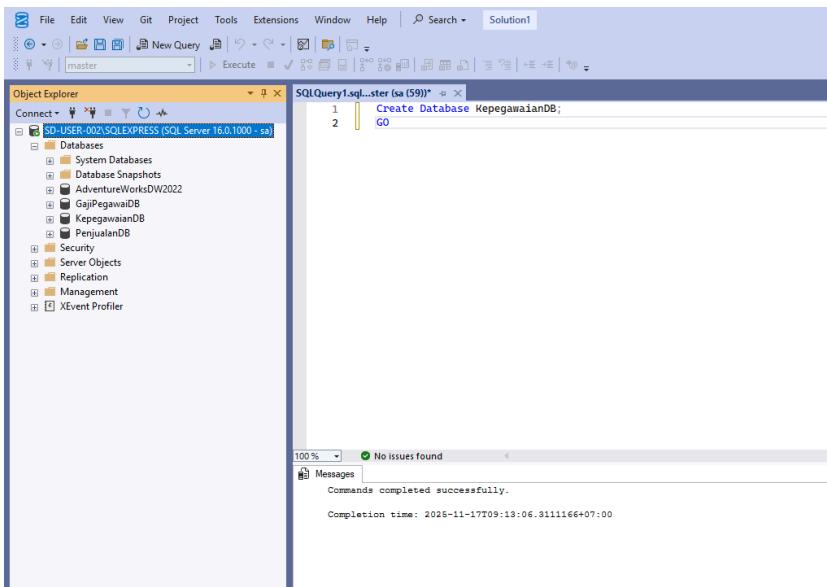
- 1. Fadhil Fitra Wijaya - 122450082**
- 2. Ali Aristo Muthahhari Parisi - 123450088**
- 3. Benget Sidabutar - 123450047**
- 4. Nobel Nizam Fathirizkii - 123450117**

Kelompok 8

Anggota

- Fadhil Fitra Wijaya - 122450082
- Benget Sidabutar - 123450047
- Nobel Nizam - 123450117
- Ali Aristo Aristo Muthahhari Parisi - 123450088

A. Database Terdeployed,



```
File Edit View Git Project Tools Extensions Window Help Search Solution1
[...]
master Execute 1 Create Database KepegawaianDB;
2 GO

Object Explorer
Connect Databases System Databases Database Snapshots AdventureWorksDW2022 GajiPegawaiDB KepegawaianDB PerjualanDB Security Server Objects Replication Management XEvent Profiler

SQLQuery1.sql...ster (sa (59))*
1 Create Database KepegawaianDB;
2 GO

100 % No issues found
Messages Commands completed successfully.
Completion time: 2025-11-17T09:13:06.3111166+07:00
```

Database sudah berhasil dijalankan dalam local, dan telah ditambahkan tabel-tabel berikut.

1. Tabel Fakta

Fact Table	Deskripsi	Sifat Fact	Sumber Entitas
Fact_Employee_Snapshot	Menyimpan status terakhir pegawai (seperti jabatan, pangkat, dan lokasi unit kerja) beserta data penggajian yang diterima pada bulan tersebut.	Transaction Fact (Mencatat setiap, status akhir pegawai)	Mengambil nominal gaji dari Gaji dan status posisi pegawai bulan tersebut dari Pegawai
Fact_Attendance	Mencatat log kehadiran harian pegawai.	Transaction Fact (Mencatat setiap event kehadiran)	Absen kehadiran

Fact_Performance	Menyimpan skor SKP periodik (Penilaian Kinerja).	Snapshot Fact (Mencatat status/hasil pengukuran pada periode tertentu)	Penilaian_Kinerja
------------------	--	--	-------------------

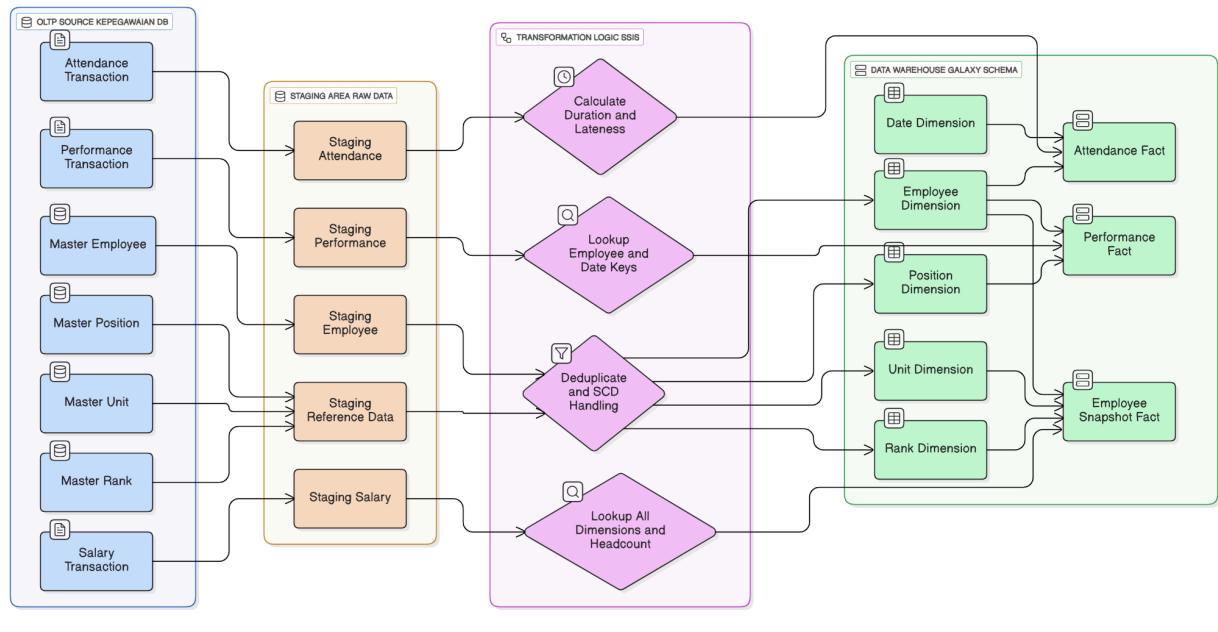
2. Tabel Dimensi

Dimension Table	Deskripsi	Sumber Entitas	Atribut Kunci
Dim_Pegawai	Data master pegawai (Dosen/Tendik) untuk mendeskripsikan "siapa".	Pegawai	NIP, Nama, Tgl_Lahir, Jenis_Kelamin
Dim_Unit	Struktur organisasi/tempat pegawai berada ("di mana").	Unit Kerja	Id_Unit, Nama Unit, Fakultas
Dim_Jabatan	Data master jabatan pegawai ("apa" posisi mereka).	Jabatan	Id_Jabatan, Nama_Jabatan
Dim_Pangkat	Data master pangkat/golongan pegawai.	Pangkat	Id_Pangkat, Golongan
Dim_Date	Data Tahunan/Bulanan/Harian	Date	DateKey, Year, Month, Day

B. ETL Arsitektur

Dokumentasi Teknis: ETL Master Load

1. Komponen & Script yang Digunakan



eraser

Script ini ditulis menggunakan bahasa T-SQL (Transact-SQL) yang spesifik untuk lingkungan Microsoft SQL Server.

Berikut adalah rincian perintah dan logika kunci yang digunakan dalam *script*:

- Control Flow:
 - CREATE OR ALTER PROCEDURE: Membungkus seluruh logika dalam satu objek yang mudah dieksekusi ulang atau diperbarui.
 - SET NOCOUNT ON: Mengoptimalkan kinerja jaringan dengan mencegah pengiriman pesan "x rows affected" yang tidak perlu.
 - DECLARE ... GETDATE(): Digunakan untuk *logging* durasi proses.
- Data Manipulation (DML):
 - TRUNCATE TABLE: Membersihkan tabel *Staging* dengan cepat sebelum data baru dimuat (Reset log).
 - INSERT INTO ... SELECT: Memindahkan data secara massal (*bulk insert*).
 - MERGE: Digunakan pada Dim_Employee untuk menangani logika *Upsert* (Update jika ada, Insert jika baru).

- DELETE: Menghapus data spesifik pada Fact Table sebelum insert ulang untuk mencegah duplikasi.
- Transformasi & Fungsi:
 - DATEDIFF: Menghitung durasi kerja dan keterlambatan (dalam menit).
 - CAST/CONVERT: Mengubah format tanggal menjadi DateKey (format integer YYYYMMDD).
 - ISNULL: Menangani nilai *NULL* dengan memberikan nilai *default* (biasanya -1 untuk *Unknown Member*).
 - CASE WHEN: Logika bisnis untuk menentukan apakah pegawai terlambat (setelah 07:30).

2. Arsitektur Data

Arsitektur yang diterapkan dalam script ini adalah pola ETL (Extract, Transform, Load) berbasis Database. Proses transformasi berat dilakukan di dalam *Data Warehouse* itu sendiri, bukan di server aplikasi terpisah.

Alur Data (Data Flow):

1. Source (OLTP): Data transaksional mentah (tabel Ms... dan Tr...).
2. Staging Area: Tempat pendaratan sementara (Staging.MsPegawai, dll). Data di sini adalah salinan persis dari Source tetapi di-*truncate* setiap kali proses berjalan.
3. Data Warehouse (OLAP): Tujuan akhir dengan skema Galaxy Schema.
 - Dimensions: Tabel referensi (Dim_Unit, Dim_Employee).
 - Facts: Tabel transaksi bisnis yang berisi metrik (Fact_Attendance, Fact_Performance, Fact_Employee_Snapshot).

3. Tools & Lingkungan

- Database Engine: Microsoft SQL Server.
- Orchestrator: SQL Stored Procedure (ETL_Master_Load). Prosedur ini bertindak sebagai *container* tunggal yang menjalankan proses secara berurutan (sekuensial).
- Scheduling (Implied): Biasanya prosedur ini akan dipanggil oleh SQL Server Agent atau *external scheduler* (seperti SSIS atau Airflow) secara harian (Daily Batch).

4. Strategi ETL

Strategi ini dirancang untuk menangani data historis, perubahan data master, dan kinerja.

A. Strategi Staging: *Full Load (Truncate & Load)*

- Metode: Setiap kali ETL berjalan, tabel staging dikosongkan (TRUNCATE) lalu diisi ulang sepenuhnya dari sumber.
- Tujuan: Memastikan data di staging selalu sinkron dengan sumber tanpa perlu memikirkan *change tracking* yang rumit di sisi sumber.

B. Strategi Dimensi (Dimensions)

1. Dim_Unit, Dim_Position, Dim_Rank:
 - Metode: *Incremental Insert*.
 - Logika: Menggunakan WHERE NOT EXISTS. Hanya data referensi baru yang dimasukkan. Data lama tidak diubah.
2. Dim_Employee:
 - Metode: SCD Type 1 (Slowly Changing Dimension Type 1).
 - Logika: Menggunakan MERGE. Jika data pegawai berubah (misal ganti nama atau status), data lama di-*overwrite*. Tidak ada riwayat perubahan atribut dimensi yang disimpan.

C. Strategi Fakta (Facts)

1. Fact_Attendance (Transaksi Harian):
 - Metode: Idempotency (Delete-Insert).
 - Logika: Sebelum insert, script menghapus data pada DateKey yang bersangkutan. Ini mencegah duplikasi jika ETL dijalankan ulang (Rerun) pada hari yang sama.
 - Transformasi: Menghitung DurasiKerja dan MenitTerlambat *on-the-fly* saat pemindahan dari Staging ke Fact.
2. Fact_Performance (Transaksi Berkala):
 - Metode: *Check Existence*.
 - Logika: Data hanya dimasukkan jika kombinasi DateKey dan EmployeeKey belum ada.
3. Fact_EmployeeSnapshot (Snapshot Bulanan):
 - Metode: Periodic Snapshot Fact Table.
 - Logika: Mengambil potret gaji dan posisi pegawai pada satu titik waktu (awal bulan). DateKey dibentuk dari Tahun + Bulan + '01'. Ini berguna untuk analisis tren gaji bulanan.

D. Penanganan Kualitas Data

- Surrogate Key Lookup: Menggunakan LEFT JOIN ke tabel Dimensi untuk mendapatkan Key (bukan ID asli).
- Handling Null: Menggunakan ISNULL(Key, -1) untuk memastikan integritas referensial tetap terjaga meskipun data dimensi belum lengkap (*Late Arriving Dimension*).

C. PENERAPAN SECURITY

1. Pembuatan Roles

```
SQLQuery1.sq...Fadhil (91)*  ↗ X
1 USE Kepegawaian_DB;
2 GO
3
4 -- 1. Membuat Login & User Baru (Simulasi User Kampus)
5 -- Login ini bernama 'User_Itera' dengan password 'Mahasiswa123!'
6 CREATE LOGIN User_Itera WITH PASSWORD = 'Mahasiswa123!';
7 | CREATE USER User_Itera FOR LOGIN User_Itera;
8 | GO
9
10 -- 2. Membuat Role Khusus (Role: ReportViewer)
11 CREATE ROLE ReportViewer;
12 GO
13
14 -- 3. Memberi Izin (Permissions)
15 -- Hanya boleh SELECT (Baca) di semua tabel schema 'dbo'
16 GRANT SELECT ON SCHEMA::dbo TO ReportViewer;
17 | -- Tolak izin DELETE/UPDATE (Opsional, untuk penegasan)
18 | DENY DELETE, UPDATE, INSERT ON SCHEMA::dbo TO ReportViewer;
19 | GO
20
21 -- 4. Memasukkan User ke dalam Role
22 ALTER ROLE ReportViewer ADD MEMBER User_Itera;
23 GO

100 %  No issues found
Messages  Commands completed successfully.

Completion time: 2025-12-14T14:15:38.3388047+07:00
```

Pembuatan Roles dengan menggunakan User = User_Itera, dan Password = Mahasiswa123!.

2. Penerapan Masking untuk data yang penting (Contoh: GajiPokok)

```
SQLQuery1.sq...Fadhil (91)*  ↗ X
1 USE Kepegawaian_DB;
2 GO
3
4 -- Menerapkan Masking pada kolom GajiPokok
5 -- 'default()' akan mengubah angka menjadi 0 bagi user yang tidak berhak
6 ALTER TABLE Fact_EmployeeSnapshot
7 | ALTER COLUMN GajiPokok ADD MASKED WITH (FUNCTION = 'default()');
8 | GO
9
10 -- Opsional: Masking Email/NIP (menjadi xxxx@itera...)
11 | -- ALTER TABLE Dim_Employee
12 | -- ALTER COLUMN NIP ADD MASKED WITH (FUNCTION = 'partial(1, "xxx", 0')';

100 %  No issues found
Messages  Commands completed successfully.

Completion time: 2025-12-14T14:17:01.0813608+07:00
```

3. Tabel Audit Log

The screenshot shows a SQL query window titled "SQLQuery1.sq...Fadhil (91)*". The code creates a table "Audit_Log_Kepergawaiian" and a trigger "TR_Audit_Gaji_Change". The table has columns: LogID (primary key, identity), WaktuKejadian (datetime default GETDATE()), UserPelaku (varchar(100)), Aksi (varchar(50)), and Keterangan (varchar(255)). The trigger "TR_Audit_Gaji_Change" is created on the "Fact_EmployeeSnapshot" table, fired after an update, and inserts data into the audit log table. The audit log entry includes the user's name, update type ('UPDATE'), and a descriptive message about the change being detected in the fact table.

```
1 USE Kepergawaiian_DB;
2 GO
3
4 -- 1. Buat Tabel Penampung Log
5 CREATE TABLE Audit_Log_Kepergawaiian (
6     LogID INT IDENTITY(1,1) PRIMARY KEY,
7     WaktuKejadian DATETIME DEFAULT GETDATE(),
8     UserPelaku VARCHAR(100),
9     Aksi VARCHAR(50),
10    Keterangan VARCHAR(255)
11 );
12 GO
13
14 -- 2. Buat Trigger (CCTV Otomatis)
15 CREATE OR ALTER TRIGGER TR_Audit_Gaji_Change
16 ON Fact_EmployeeSnapshot
17 AFTER UPDATE
18 AS
19 BEGIN
20     SET NOCOUNT ON;
21
22     -- Masukkan data ke log setiap ada update
23     INSERT INTO Audit_Log_Kepergawaiian (UserPelaku, Aksi, Keterangan)
24     SELECT
25         SUSER_NAME(), -- Nama User Login
26         'UPDATE',
27         'Perubahan data terdeteksi pada tabel Fact_EmployeeSnapshot'
28     END;
29 GO
```

100 % No issues found

Messages Commands completed successfully.

Completion time: 2025-12-14T14:17:40.5497908+07:00

Menampung segala aktivitas perubahan data

4. Testing

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top window is titled "SQLQuery1.sq...adhil (105)*". It contains the following T-SQL script:

```
1 USE Kepegawaian_DB;
2 GO
3
4 /* PRINT '*** BUKTI 1: ADMIN MELIHAT DATA ASLI ***';
5 -- Ini dijalankan sebagai Fadhil (Admin)
6 */
7 SELECT TOP 5 EmployeeKey, GajiPokok
8 FROM Fact_EmployeeSnapshot;
9
10 PRINT '';
11 PRINT '*** BUKTI 2: USER BIASA TERKENA MASKING ***';
12 -- Perintah sakti untuk menyamar jadi User_Itera
13 EXECUTE AS USER = 'User_Itera';
14
15 -- Jalankan query yang sama persis
16 SELECT TOP 5 EmployeeKey, GajiPokok
17 FROM Fact_EmployeeSnapshot;
18
19 -- Kembali menjadi Admin
20 REVERT;
21 GO
```

The bottom window is titled "Results" and displays two result sets:

	EmployeeKey	GajiPokok
1	1	5000000.00
2	2	3500000.00
3	3	4000000.00
4	4	3000000.00
5	5	5500000.00

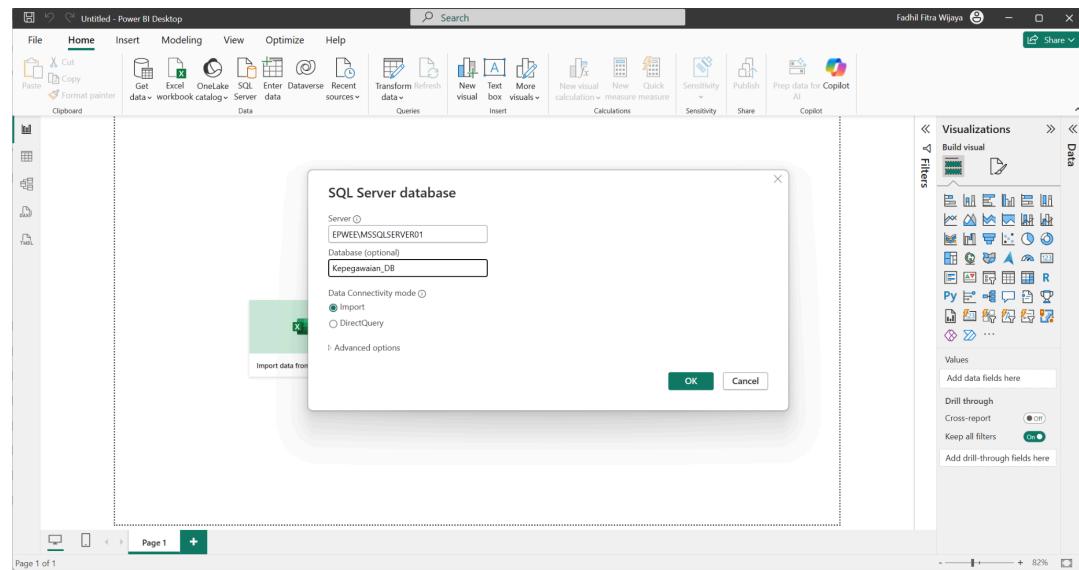
	EmployeeKey	GajiPokok
1	1	0.00
2	2	0.00
3	3	0.00
4	4	0.00
5	5	0.00

A status bar at the bottom indicates: "Query executed successfully."

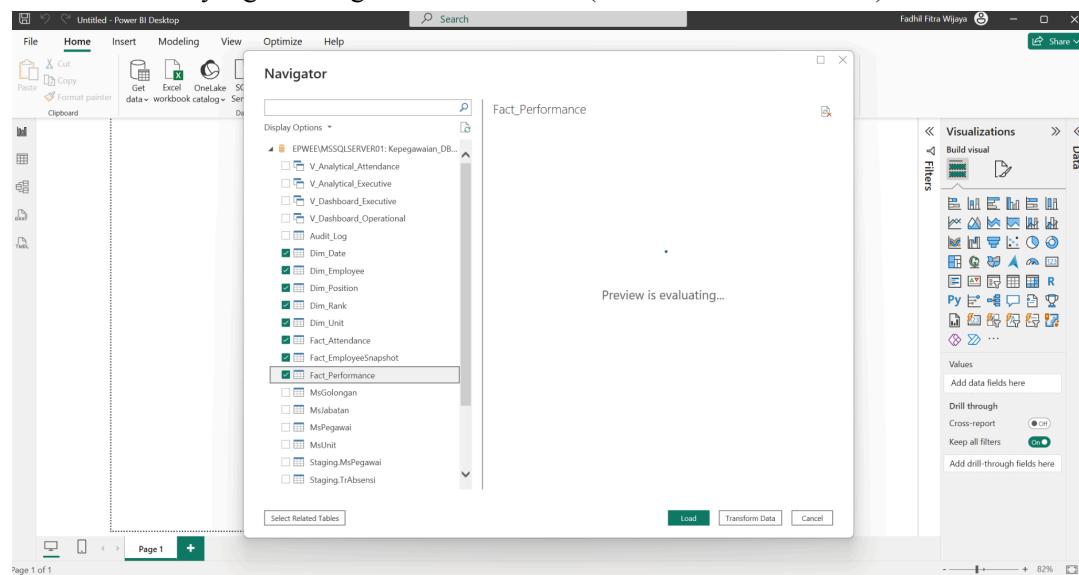
Memperlihatkan perbedaan saat user dan admin yang mengakses database Atas, admin akses database dan bawah, user akses database.

D. CONNECT POWERBI

E. Melakukan Connection PowerBI dari Database Kepgawai

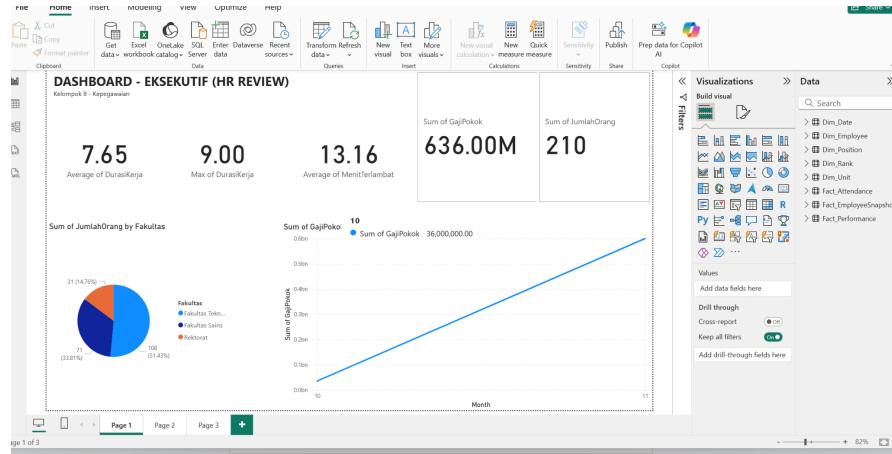


F. Memilih Kolom attribut yang akan digunakan visualisasi (Tabel Dimensi dan Fact)



G. Pembuatan Dashboard

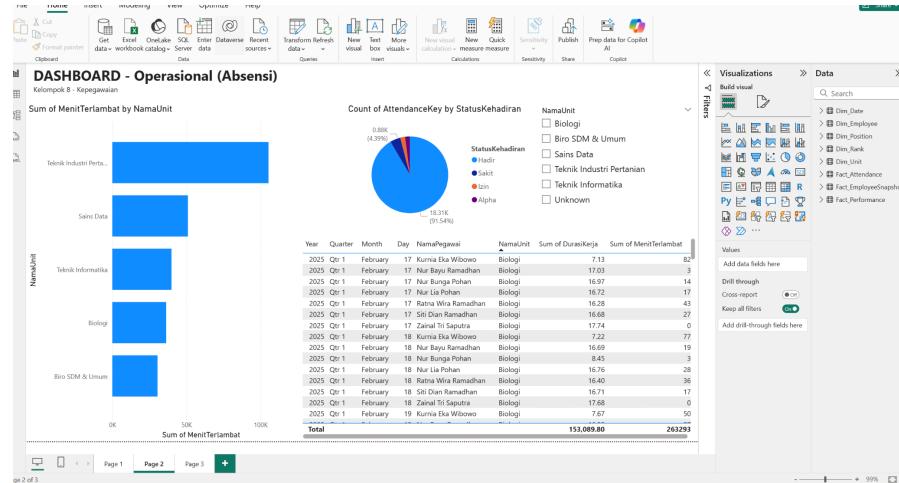
a. Dashboard Eksekutif (HR REVIEW)



Dashboard ini berfungsi sebagai ringkasan strategis bagi pimpinan perusahaan atau manajer HR untuk melihat kondisi umum tenaga kerja dan beban biaya secara cepat (helikopter view).

- H. Ringkasan Biaya & Demografi:** Menampilkan angka kunci seperti total anggaran gaji pokok (636.00M) dan jumlah total karyawan (210 orang).
- I. Retensi Karyawan:** Memantau rata-rata lama bekerja (7.65 tahun) untuk melihat loyalitas karyawan.
- J. Sebaran & Tren:** Visualisasi grafik garis menunjukkan tren pengeluaran gaji dari waktu ke waktu, serta diagram lingkaran untuk melihat proporsi jumlah karyawan per fakultas/divisi.

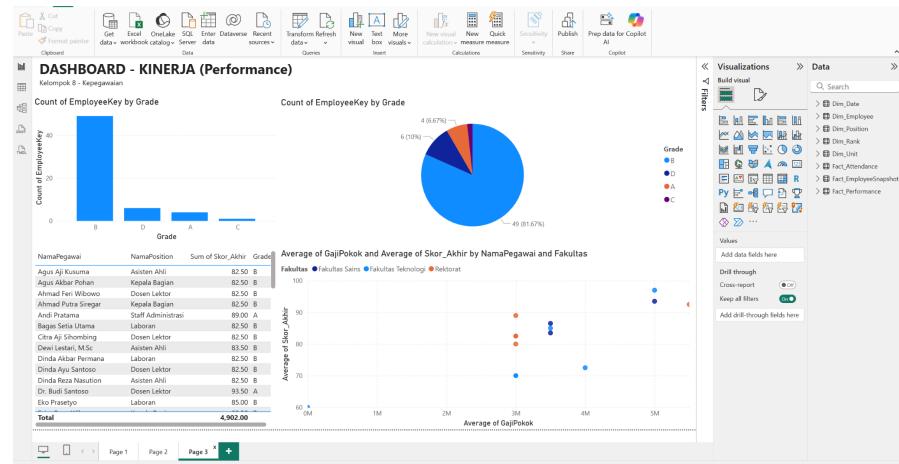
a. Dashboard Operasional (Absensi)



Halaman ini dirancang untuk pemantauan operasional harian, khususnya yang berkaitan dengan kedisiplinan dan tingkat kehadiran karyawan di berbagai unit kerja.

- K. **Analisis Keterlambatan:** Grafik batang menunjukkan akumulasi menit keterlambatan yang dikelompokkan berdasarkan unit kerja, memudahkan identifikasi departemen mana yang paling tidak disiplin.
- L. **Status Kehadiran:** Diagram lingkaran memetakan proporsi karyawan yang Hadir, Sakit, Izin, atau Alpha.
- M. **Detail Karyawan:** Tabel di bagian bawah memberikan rincian granular per individu, sehingga manajemen bisa menelusuri siapa saja yang sering terlambat berserta durasi keterlambatannya.

a. Dashboard Kinerja (Performance)



Dashboard ini fokus pada analisis kualitas SDM, struktur kepangkatan (grading), dan hubungannya dengan kompensasi yang diberikan.

- **Distribusi Pangkat (Grade):** Grafik batang memvisualisasikan jumlah karyawan yang berada di setiap level atau *grade*, memberikan gambaran tentang struktur organisasi (apakah gemuk di bawah atau merata).
- **Korelasi Gaji vs Kinerja:** *Scatter plot* (grafik titik) digunakan untuk melihat hubungan antara besaran gaji pokok dengan skor akhir kinerja. Ini membantu mendeteksi apakah karyawan bergaji tinggi sudah memberikan kinerja yang sepadan.
- **Rincian Penilaian:** Tabel menampilkan daftar nama pegawai lengkap dengan jabatan, gaji, dan skor akhir mereka untuk evaluasi individu.