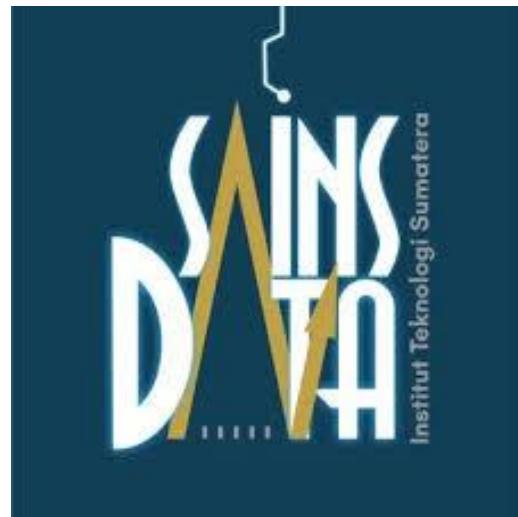


LAPORAN TUGAS BESAR

PERGUDANGAN DATA

MISI 2



Oleh :

Kelompok 10

- 1. Fadhil Fitra Wijaya - 122450082**
- 2. Ali Aristo Muthahhari Parisi - 123450088**
- 3. Benget Sidabutar - 123450047**
- 4. Nobel Nizam Fathirizkii - 123450117**
- 5.**

Misi 2 Pergudangan Data

Kelompok 8

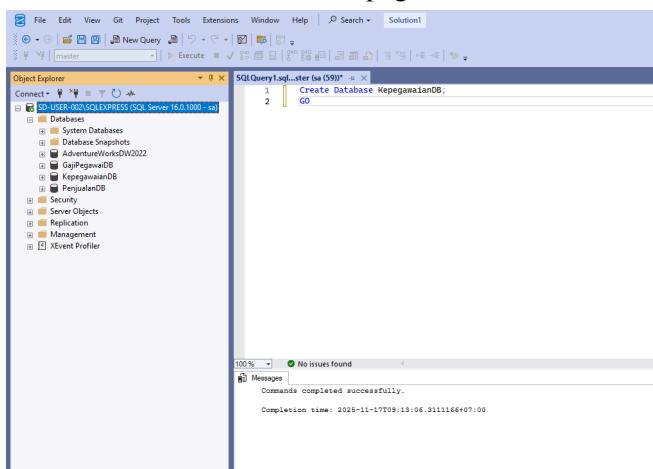
Anggota

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

Progres Misi 2

A. DATABASE KEPEGAWAIAN

1. Membuat Database Kepegawaian



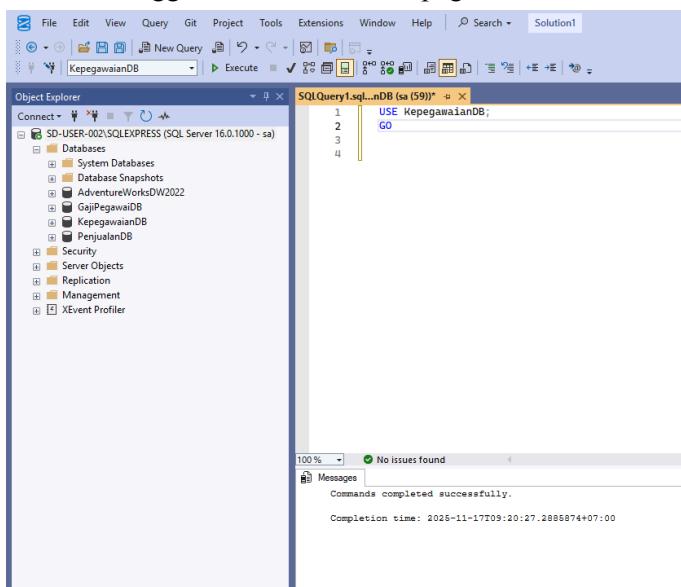
The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the connection to "SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa)" and lists databases like "AdventureworksDW2022", "GajiPegawaiDB", "KepegawaianDB", and "PenjualanDB".
- SQL Query Editor:** Contains the following T-SQL code:

```
1 Create Database KepegawaianDB;
2 GO;
```
- Messages Window:** Displays the output:

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

2. Menggunakan Database Kepegawaian



The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the connection to "SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa)" and lists databases like "AdventureworksDW2022", "GajiPegawaiDB", "KepegawaianDB", and "PenjualanDB".
- SQL Query Editor:** Contains the following T-SQL code:

```
1 USE KepegawaianDB;
2 GO;
```
- Messages Window:** Displays the output:

```
No issues found
Commands completed successfully.
Completion time: 2025-11-17T09:20:27.2085874+07:00
```

3. Membuat Dimensi Tabel

- Tabel 1 (Dim_Date)

The screenshot shows the Object Explorer on the left with the connection to 'KepgawaiandB'. The 'Tables' node under 'KepgawaiandB' is expanded. The 'Script' button is selected for the 'Dim_Date' table. The 'Script' pane displays the T-SQL code for creating the table:

```
1  CREATE TABLE [dbo].[Dim_Date] (
2      DateKey INT PRIMARY KEY, -- YYYYMMDD
3      FullDate DATE NOT NULL,
4      DayOfWeek TINYINT,
5      DayName VARCHAR(10),
6      DayMonth TINYINT,
7      WeekOfYear TINYINT,
8      MonthNumber TINYINT,
9      MonthName VARCHAR(10),
10     Quarter TINYINT,
11     Year SMALLINT,
12     IsWeekend BIT,
13     IsHoliday BIT,
14     HolidayName VARCHAR(100)
15 );
16 GO
17
```

The status bar at the bottom right indicates a completion time of 2026-11-17T09:22:24.6928138+07:00.

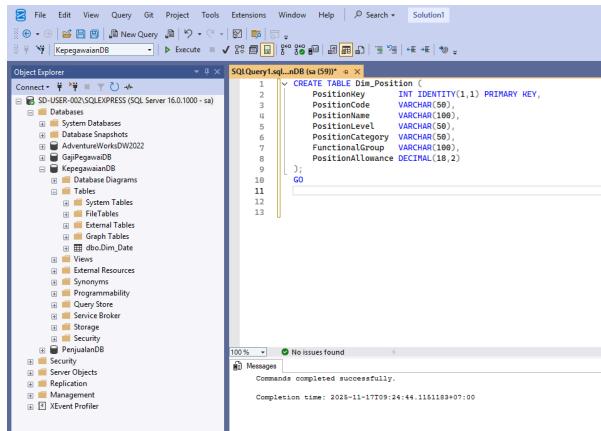
- Tabel 2 (Dim_Employee)

The screenshot shows the Object Explorer on the left with the connection to 'KepgawaiandB'. The 'Tables' node under 'KepgawaiandB' is expanded. The 'Script' button is selected for the 'Dim_Employee' table. The 'Script' pane displays the T-SQL code for creating the table:

```
1  CREATE TABLE [dbo].[Dim_Employee] (
2      EmployeeKey INT IDENTITY(1,1) PRIMARY KEY,
3      NIP VARCHAR(50) NOT NULL,
4      EmployeeName VARCHAR(100),
5      Birthdate DATE,
6      Gender CHAR(1),
7      EmployeeType VARCHAR(50),
8      EmploymentStatus VARCHAR(50),
9      HireDate DATE,
10    TerminationDate DATE,
11    HighestEducation VARCHAR(50),
12    AddressFields
13    IsCurrent BIT DEFAULT 1,
14    EffectiveDate DATE,
15    ExpiryDate DATE,
16    CONSTRAINT UQ_Employee_NIP_Effective UNIQUE (NIP, EffectiveDate)
17 );
18
19
20
21
22
```

The status bar at the bottom right indicates a completion time of 2026-11-17T09:24:15.1187604+07:00.

- Tabel 3 (Dim_Position)

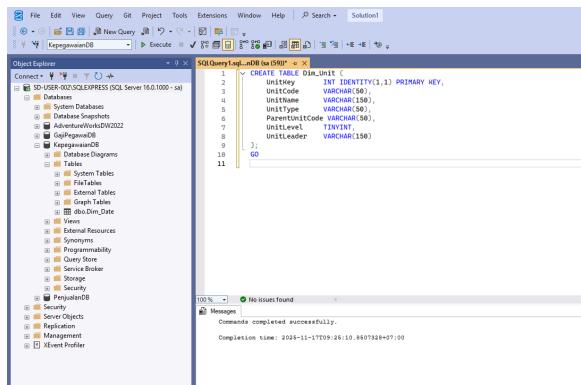


The screenshot shows the Object Explorer on the left with the connection to 'SD-USR-003-SQLEXPRESS (SQL Server 16.0.1000 - sa)'. The 'Tables' node under 'AdventureWorksDW' is expanded. A new query window titled 'SQLQuery1 - [master] (DTS Import)' is open, displaying the following T-SQL code:

```
CREATE TABLE Dim_Position (
    PositionKey INT IDENTITY(1,1) PRIMARY KEY,
    PositionCode VARCHAR(50),
    PositionName VARCHAR(100),
    PositionLevel VARCHAR(50),
    PositionCategory VARCHAR(50),
    FunctionalGroup VARCHAR(100),
    PositionAllowance DECIMAL(18,2)
);
```

The code is highlighted with syntax coloring. The status bar at the bottom right indicates 'Commands completed successfully.' and 'Completion time: 2024-11-17T09:24:44.1151183+07:00'.

- Tabel 4 (Dim_Unit)



The screenshot shows the Object Explorer on the left with the same connection. The 'Tables' node under 'AdventureWorksDW' is expanded. A new query window titled 'SQLQuery2 - [master] (DTS Import)' is open, displaying the following T-SQL code:

```
CREATE TABLE Dim_Unit (
    UnitKey INT IDENTITY(1,1) PRIMARY KEY,
    UnitCode VARCHAR(50),
    UnitName VARCHAR(150),
    UnitType VARCHAR(50),
    PositionCode INT,
    UnitLevel1 TINYINT,
    UnitLeader VARCHAR(150)
);
```

The code is highlighted with syntax coloring. The status bar at the bottom right indicates 'Commands completed successfully.' and 'Completion time: 2024-11-17T09:26:10.8807328+07:00'.

- Tabel 5 (Dim_Rank)

The screenshot shows the SQL Server Management Studio (SSMS) interface. In the Object Explorer, the database 'KepengawaiandB' is selected. In the center pane, a query window titled 'SQLQuery1.sdlDB (sa (S9))' contains the following SQL code:

```
CREATE TABLE Dim_Rank (
    RankKey INT IDENTITY(1,1) PRIMARY KEY,
    RankCode VARCHAR(50),
    RankName VARCHAR(100),
    Grade VARCHAR(5),
    Class VARCHAR(5),
    BaseSalaryMin DECIMAL(18, 2),
    BaseSalaryMax DECIMAL(18, 2)
);
GO
```

In the bottom right corner of the SSMS interface, there is a message: 'Commands completed successfully.' and 'Completion time: 2025-11-17T09:25:34.0996946+07:00'.

- Tabel 6 (Dim_LeaveType)

The screenshot shows the SQL Server Management Studio (SSMS) interface. In the Object Explorer, the database 'KepengawaiandB' is selected. In the center pane, a query window titled 'SQLQuery1.sdlDB (sa (S9))' contains the following SQL code:

```
CREATE TABLE Dim_LeaveType (
    LeaveTypeKey INT IDENTITY(1,1) PRIMARY KEY,
    LeaveTypeCode VARCHAR(50),
    LeaveTypeName VARCHAR(100),
    LeaveCategory VARCHAR(50),
    MaxDaysPerYear INT
);
GO
```

In the bottom right corner of the SSMS interface, there is a message: 'Commands completed successfully.' and 'Completion time: 2025-11-17T09:26:11.6751627+07:00'.

- Tabel 7 (Dim_Training Program)

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure for "KepegawaianDB".
- SQL Query1 (sa@(59)):** A query window containing T-SQL code to create a table named "Dim_TrainingProgram". The table has columns: TrainingProgramKey (INT, primary key, identity), TrainingCode (VARCHAR(50)), TrainingName (VARCHAR(150)), TrainingCategory (VARCHAR(100)), TrainingType (VARCHAR(50)), and DurationHours (INT). The code ends with a GO statement.
- Messages:** A status bar at the bottom indicating "Commands completed successfully." and the completion time: "Completion time: 2025-11-17T09:26:43.1146918+07:00".

4. Tabel Fact

- Fact_Employee_Snapshot

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure:
 - Connect to: SD-USER-002-SQLEXPRESS (SQL Server 16.0.1000 - sa)
 - Databases:
 - System Databases
 - Database Snapshots
 - AdventureworksDW2022
 - GajiPegawaiDB
 - KepengawaiandB
 - Tables:
 - System Tables
 - External Tables
 - Graph Tables
 - dbo.Dim_Date
 - Views
 - External Resources
 - Synonyms
 - Procedures
 - Policy Store
 - Service Broker
 - Storage
 - Security
 - PergulanganDB
 - Security
 - Server Objects
 - Replication
 - Management
 - VS-Client Profiler
- SQLQuery1.sql.nDB (a (59))**: A query window containing T-SQL code to create a fact table with various columns and constraints.
- Messages:** Shows the message "Commands completed successfully."
- Completion time:** 2028-11-17T09:34:10.8004919+07:00

- Fact Attendance

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The Object Explorer on the left lists the database 'KepgawaiandB' under the connection 'SD-USER-002\SQL EXPRESS (SQL Server 16.0.1000 - sa)'. The query editor window on the right contains the following T-SQL script:

```
CREATE TABLE Fact_Attendance (
    AttendanceKey INT IDENTITY(1,1) PRIMARY KEY,
    Datekey INT NOT NULL,
    Employeekey INT NOT NULL,
    Unitkey INT,
    -- Measures
    CheckInTime DATETIME,
    CheckOutTime DATETIME,
    WorkingHours DECIMAL(5,2),
    LateMinutes INT,
    OvertimeHours DECIMAL(5,2),
    IsPresent BIT,
    IsLate BIT,
    FOREIGN KEY (Datekey) REFERENCES Dim_Dates(DateKey),
    FOREIGN KEY (Employeekey) REFERENCES Dim_Employee(EmployeeKey),
    FOREIGN KEY (Unitkey) REFERENCES Dim_Unit(UnitKey)
);
GO
```

- Fact Performance

The screenshot shows the SQL Server Management Studio (SSMS) interface. The Object Explorer on the left lists databases, tables, and other objects under the connection 'SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa)'. The 'KepgawainanDB' database is selected. The 'Tables' node under it shows 'Fact_Performance'. The 'Script' tab of the 'Fact_Performance' table is open, displaying the CREATE TABLE script. The script creates the table with columns: PerformanceKey (INT IDENTITY(1,1) PRIMARY KEY), EmployeeKey (INT NOT NULL), EvaluationDateKey (INT NOT NULL), PeriodStartDateKey (INT), PeriodEndDateKey (INT), EvaluatorKey (INT, -- FK Dim_Employee juga), SHPScore (DECIMAL(5,2)), BehaviorScore (DECIMAL(5,2)), TotalScore (DECIMAL(5,2)), TargetAchievement (DECIMAL(5,2)), PerformanceRating (VARCHAR(50)), and IsPromotionEligible (BIT). It also includes FOREIGN KEY constraints linking EmployeeKey to Dim_Employee(EmployeeKey), EvaluationDateKey to Dim_Employee(EvaluationDateKey), PeriodStartDateKey to Dim_Date(DateKey), and PeriodEndDateKey to Dim_Date(DateKey). The 'Messages' pane at the bottom shows 'Commands completed successfully.' and the completion time '2025-11-17T09:37:10.4760024+07:00'.

```
CREATE TABLE Fact_Performance (
    PerformanceKey INT IDENTITY(1,1) PRIMARY KEY,
    EmployeeKey INT NOT NULL,
    EvaluationDateKey INT NOT NULL,
    PeriodStartDateKey INT,
    PeriodEndDateKey INT,
    EvaluatorKey INT, -- FK Dim_Employee juga
    SHPScore DECIMAL(5,2),
    BehaviorScore DECIMAL(5,2),
    TotalScore DECIMAL(5,2),
    TargetAchievement DECIMAL(5,2),
    PerformanceRating VARCHAR(50),
    IsPromotionEligible BIT,
    FOREIGN KEY (EmployeeKey) REFERENCES Dim_Employee(EmployeeKey),
    FOREIGN KEY (EvaluationDateKey) REFERENCES Dim_Employee(EvaluationDateKey),
    FOREIGN KEY (PeriodStartDateKey) REFERENCES Dim_Date(DateKey),
    FOREIGN KEY (PeriodEndDateKey) REFERENCES Dim_Date(DateKey)
);
GO
```

B. MELAKUKAN INDEXES

1. INDEXES FOR KepegawaianDB

The screenshot shows the SSMS interface. The Object Explorer on the left lists databases: SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa), GajiPegawaiDB, and KepegawaianDB. The KepegawaianDB node is expanded, showing its structure: Databases, Tables (including System Tables, FileTables, External Tables, Graph Tables, and dbo.Dim_Date), Views, External Resources, Synonyms, Programmability, Query Store, Service Broker, Storage, Security, and PenjualanDB. The right pane contains a query window titled 'SQLQuery1.sql...nDB (sa (59))'. It has three lines of code:

```
1 USE KepegawaianDB;
2 GO
3
```

Below the query window is the 'Messages' pane, which displays the message 'Commands completed successfully.' and the completion time 'Completion time: 2025-11-17T09:38:56.4351474+07:00'.

A. INDEX UNTUK DIMENSION TABLES

- Dim_Date

The screenshot shows the SSMS interface. The Object Explorer on the left lists databases: SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa), GajiPegawaiDB, and KepegawaianDB. The KepegawaianDB node is expanded, showing its structure: Databases, Tables (including System Tables, FileTables, External Tables, Graph Tables, and dbo.Dim_Date), Views, External Resources, Synonyms, Programmability, Query Store, Service Broker, Storage, Security, and PenjualanDB. The right pane contains a query window titled 'SQLQuery1.sql...nDB (sa (59))'. It has three lines of code:

```
1 CREATE INDEX IX_DimDate_YearMonth ON Dim_Date(Year, MonthNumber);
2 GO
3
```

Below the query window is the 'Messages' pane, which displays the message 'Commands completed successfully.' and the completion time 'Completion time: 2025-11-17T09:41:53.4461707+07:00'.

- Dim_Employee

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'KepegawaianDB' is selected. In the center pane, a query window titled 'SQLQuery1.sql...DB (sa (59))' contains the following T-SQL code:

```
1  CREATE UNIQUE INDEX IX_DimEmployee_NIP_Effective
2      ON Dim_Employee(NIP, EffectiveDate);
3
4  CREATE INDEX IX_DimEmployee_IsCurrent ON Dim_Employee(IsCurrent);
5
6
```

The 'Messages' pane at the bottom right shows the output: 'Commands completed successfully.' and 'Completion time: 2026-11-17T09:42:41.5775736+07:00'.

- **Dim_Position**

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'KepegawaianDB' is selected. In the center pane, a query window titled 'SQLQuery1.sql...DB (sa (59))' contains the following T-SQL code:

```
1  CREATE INDEX IX_PositionCode ON Dim_Position(PositionCode);
2
3
```

The 'Messages' pane at the bottom right shows the output: 'Commands completed successfully.' and 'Completion time: 2026-11-17T09:44:39.4759993+07:00'.

- Dim_Unit

The screenshot shows the SSMS interface with the Object Explorer on the left and a query window on the right. The query window contains the following T-SQL code:

```
CREATE INDEX IX_UnitCode ON Dim_Unit(UnitCode);
CREATE INDEX IX_ParentUnitCode ON Dim_Unit(ParentUnitCode);
GO
```

The status bar at the bottom of the window indicates "Commands completed successfully." and "Completion time: 2028-11-17T09:45:07.8214543+07:00".

- Dim_Rank

The screenshot shows the SSMS interface with the Object Explorer on the left and a query window on the right. The query window contains the following T-SQL code:

```
CREATE INDEX IX_RankCode ON Dim_Rank(RankCode);
GO
```

The status bar at the bottom of the window indicates "Commands completed successfully." and "Completion time: 2028-11-17T09:45:14.8677780+07:00".

- Dim_LeaveType

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'KepegawaiandB' is selected. In the center pane, a query window titled 'SQLQuery1.sql...nDB (sa (59))' contains the following T-SQL code:

```
CREATE INDEX IX_LeaveTypeCode ON Dim_LeaveType(LeaveTypeCode);
GO
```

The 'Messages' pane at the bottom right shows the execution results:

Commands completed successfully.
Completion time: 2025-11-17T09:46:01.2671805+07:00

- Dim_TrainingProgram

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'KepegawaiandB' is selected. In the center pane, a query window titled 'SQLQuery1.sql...nDB (sa (59))' contains the following T-SQL code:

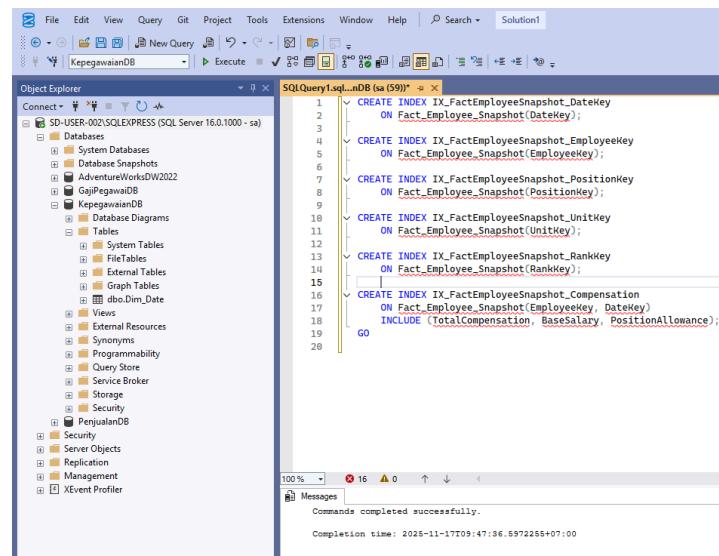
```
CREATE INDEX IX_TrainingCode ON Dim_TrainingProgram(TrainingCode);
GO
```

The 'Messages' pane at the bottom right shows the execution results:

Commands completed successfully.
Completion time: 2025-11-17T09:46:23.9627482+07:00

B. INDEX UNTUK FACT TABLES (FK + Query Fields)

- Fact_Employee_Snapshot

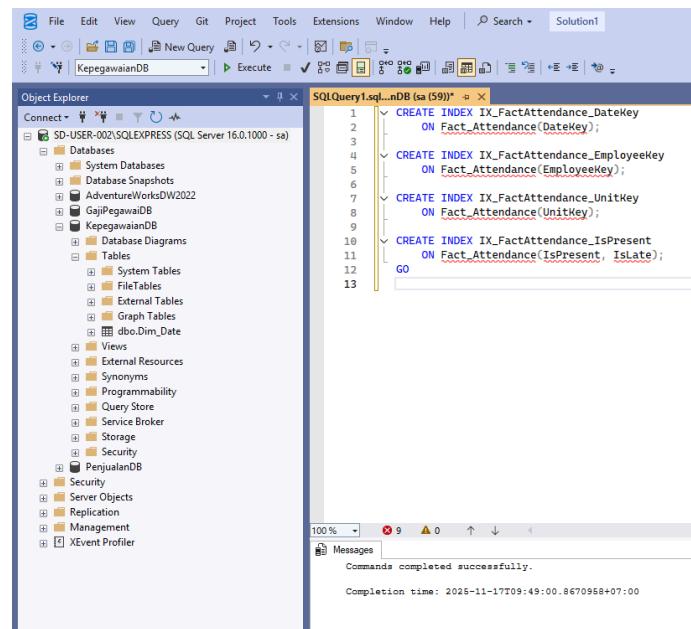


```
File Edit View Query Git Project Tools Window Help | Search Solution1
Object Explorer
Connect SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa)
Databases
System Databases
Database Snapshots
AdventureWorksDW2022
GajiPegawaiDB
KepegawaianDB
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
PenjualanDB
Security
Server Objects
Replication
Management
XEvent Profiler

SQlQuery1.sql...nDB (sa (59)) * x
CREATE INDEX IX_FactEmployeeSnapshot_DateKey
ON Fact_Employee_Snapshot(DateKey);
CREATE INDEX IX_FactEmployeeSnapshot_EmployeeKey
ON Fact_Employee_Snapshot(EmployeeKey);
CREATE INDEX IX_FactEmployeeSnapshot_PositionKey
ON Fact_Employee_Snapshot(PositionKey);
CREATE INDEX IX_FactEmployeeSnapshot_UnitKey
ON Fact_Employee_Snapshot(UnitKey);
CREATE INDEX IX_FactEmployeeSnapshot_RankKey
ON Fact_Employee_Snapshot(RankKey);
CREATE INDEX IX_FactEmployeeSnapshot_Compensation
ON Fact_Employee_Snapshot(EmployeeKey, DateKey)
INCLUDE (TotalCompensation, BaseSalary, PositionAllowance);
GO

Messages
Commands completed successfully.
Completion time: 2025-11-17T09:47:36.5972255+07:00
```

- Fact_Attendance



```
File Edit View Query Git Project Tools Window Help | Search Solution1
Object Explorer
Connect SD-USER-002\SQLEXPRESS (SQL Server 16.0.1000 - sa)
Databases
System Databases
Database Snapshots
AdventureWorksDW2022
GajiPegawaiDB
KepegawaianDB
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
PenjualanDB
Security
Server Objects
Replication
Management
XEvent Profiler

SQlQuery1.sql...nDB (sa (59)) * x
CREATE INDEX IX_FactAttendance_DateKey
ON Fact_Attendance(DateKey);
CREATE INDEX IX_FactAttendance_EmployeeKey
ON Fact_Attendance(EmployeeKey);
CREATE INDEX IX_FactAttendance_UnitKey
ON Fact_Attendance(UnitKey);
CREATE INDEX IX_FactAttendance_IsPresent
ON Fact_Attendance(IsPresent, IsLate);
GO

Messages
Commands completed successfully.
Completion time: 2025-11-17T09:49:00.8670968+07:00
```

- Fact_Performance

The screenshot shows the SSMS interface with the Object Explorer on the left and a query window on the right. The query window contains T-SQL code for creating various indexes on the Fact_Performance table:

```

1  CREATE INDEX IX_FactPerformance_EmployeeKey
2    ON Fact_Performance(EmployeeKey);
3
4  CREATE INDEX IX_FactPerformance_EvaluationDateKey
5    ON Fact_Performance(EvaluationDateKey);
6
7  CREATE INDEX IX_FactPerformance_PeriodStartDateKey
8    ON Fact_Performance(PeriodStartDateKey);
9
10 CREATE INDEX IX_FactPerformance_PeriodEndDateKey
11   ON Fact_Performance(PeriodEndDateKey);
12
13 CREATE INDEX IX_FactPerformance_EvaluatorKey
14   ON Fact_Performance(EvaluatorKey);
15
16 -- covering index
17 CREATE INDEX IX_FactPerformance_Totalscore
18   ON Fact_Performance(Totalscore)
19     INCLUDE (Totalscore, SdScore, BehaviorScore, PerformanceRating);
20
21 GO

```

The message bar at the bottom indicates "Commands completed successfully." and the completion time is shown as 2025-11-17T09:50:52.0994020+07:00.

C. MELAKUKAN PARTITIONING

a. Memuat Skema dan Fungsi Partisi

The screenshot shows the SSMS interface with the Object Explorer on the left and a query window on the right. The query window contains T-SQL code for creating a function and a partition scheme:

```

1  =====
2  -- 1. SIAPKAN SKEMA & FUNGSI PARTISI (Tahun)
3  =====
4  PRINT '>>> 1. Membuat Function & Scheme...';
5
6  -- Range: Data < 2023, 2023, 2024, 2025, > 2025
7  IF NOT EXISTS (SELECT * FROM sys.partition_functions WHERE name = 'PF_TahunKepegawaiann')
8  CREATE PARTITION FUNCTION PF_TahunKepegawaiann (INT)
9    AS RANGE RIGHT FOR VALUES (20230101, 20240101, 20250101, 20260101);
10
11 IF NOT EXISTS (SELECT * FROM sys.partition_schemes WHERE name = 'PS_TahunKepegawaiann')
12 CREATE PARTITION SCHEME PS_TahunKepegawaiann
13   AS PARTITION PF_TahunKepegawaiann
14   ALL TO ([PRIMARY]);
15
16 GO

```

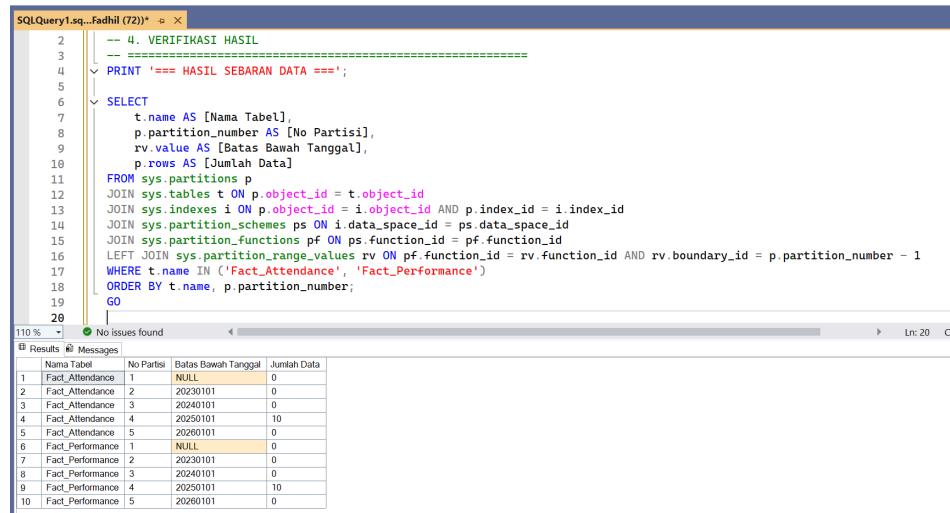
The message bar at the bottom indicates "No issues found" and "Commands completed successfully." The completion time is shown as 2025-11-24T20:18:46.0686812+07:00.

b. Partisi Tabel Fakta

```
SQLQuery1.sql...Fadhil (72)*  □ X
1  -- =====
2  -- 2. PARTISI TABEL: Fact_Attendance (Absensi)
3  -- =====
4  PRINT '">>>> 2. Mempartisi Tabel Fact_Attendance...';
5
6  DECLARE @TableName_Att NVARCHAR(100) = 'dbo.Fact_Attendance';
7  DECLARE @OldPK_Att NVARCHAR(100);
8  DECLARE @SQL_Att NVARCHAR(MAX);
9
10 -- A. Hapus Duplikat (Jaga-jaga)
11 WITH CTE_Clean AS (
12     SELECT AttendanceKey, DateKey, ROW_NUMBER() OVER (PARTITION BY AttendanceKey, DateKey ORDER BY (SELECT NULL)) AS RN
13     FROM dbo.Fact_Attendance
14 ) DELETE FROM CTE_Clean WHERE RN > 1;
15
16 -- B. Cari & Hapus PK Lama
17 SELECT TOP 1 @OldPK_Att = name FROM sys.key_constraints WHERE type = 'PK' AND parent_object_id = OBJECT_ID(@TableName_Att);
18
19 BEGIN TRY
20
21 No issues found
22 Messages
23 >>> 2. Mempartisi Tabel Fact_Attendance...
24
25 (0 rows affected)
26 -> Sukses: Fact_Attendance terpartisi.
27
28 Completion time: 2025-11-24T20:20:50.3029761+07:00
```

```
SQLQuery1.sql...Fadhil (72)*  □ X
1  -- =====
2  -- 3. PARTISI TABEL: Fact_Performance (Kinerja)
3  -- =====
4  PRINT '">>>> 3. Mempartisi Tabel Fact_Performance...';
5
6  DECLARE @TableName_Perf NVARCHAR(100) = 'dbo.Fact_Performance';
7  DECLARE @OldPK_Perf NVARCHAR(100);
8  DECLARE @SQL_Perf NVARCHAR(MAX);
9
10 -- A. Hapus Duplikat
11 WITH CTE_Clean AS (
12     SELECT PerformanceKey, DateKey, ROW_NUMBER() OVER (PARTITION BY PerformanceKey, DateKey ORDER BY (SELECT NULL)) AS RN
13     FROM dbo.Fact_Performance
14 ) DELETE FROM CTE_Clean WHERE RN > 1;
15
16 -- B. Cari & Hapus PK Lama
17 SELECT TOP 1 @OldPK_Perf = name FROM sys.key_constraints WHERE type = 'PK' AND parent_object_id = OBJECT_ID(@TableName_Perf);
18
19 BEGIN TRY
20
21 No issues found
22 Messages
23 >>> 3. Mempartisi Tabel Fact_Performance...
24
25 (0 rows affected)
26 -> Sukses: Fact_Performance terpartisi.
27
28 Completion time: 2025-11-24T20:21:56.4551097+07:00
```

c. Cek Validasi

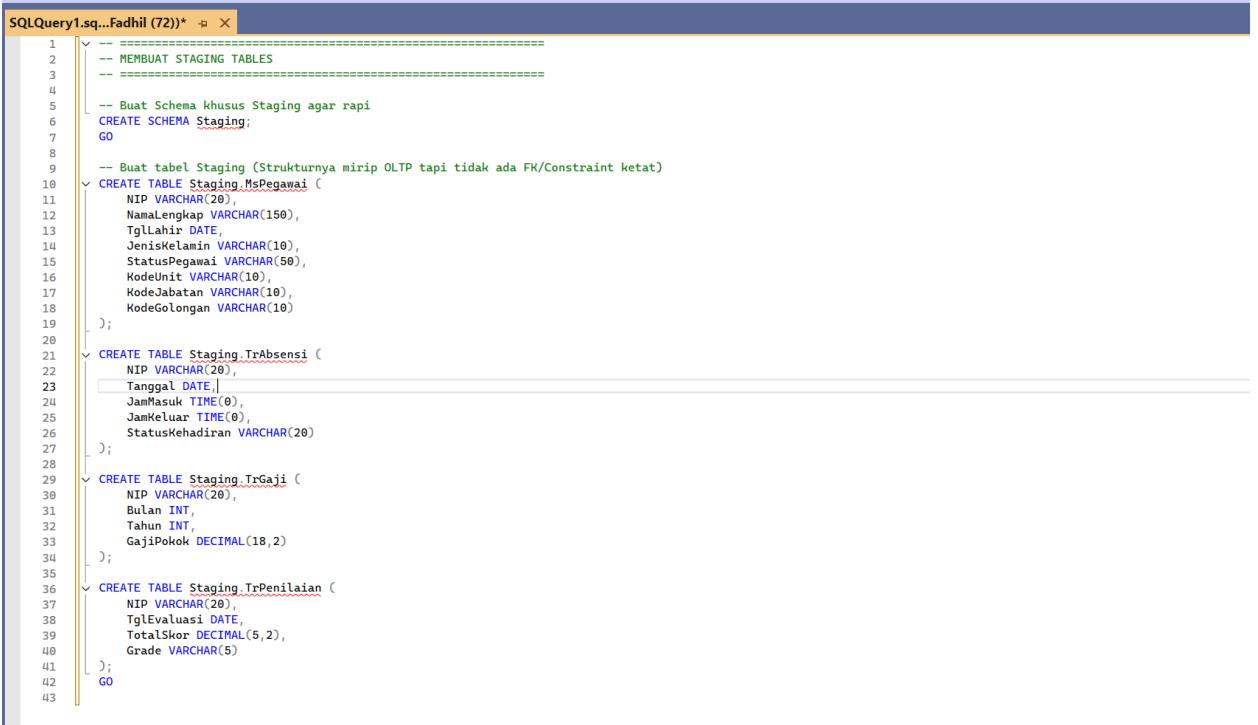


The screenshot shows a SQL Server Management Studio window with a query editor and a results grid. The query is titled '4. VERIFIKASI HASIL' and prints a message 'HASIL SEBARAN DATA'. It then performs a SELECT operation on sys.partitions and joins it with sys.tables, sys.indexes, sys.partition_schemes, and sys.partition_functions to get partition details and count rows. The WHERE clause filters for Fact_Attendance and Fact_Performance. The results grid shows 10 rows of data with columns: Nama Tabel, No Partisi, Batas Bawah Tanggal, and Jumlah Data. Most rows have 'NULL' in the 'Batas Bawah Tanggal' column and '0' in the 'Jumlah Data' column.

Nama Tabel	No Partisi	Batas Bawah Tanggal	Jumlah Data
Fact_Attendance	1	NULL	0
Fact_Attendance	2	20230101	0
Fact_Attendance	3	20240101	0
Fact_Attendance	4	20250101	10
Fact_Attendance	5	20260101	0
Fact_Performance	1	NULL	0
Fact_Performance	2	20230101	0
Fact_Performance	3	20240101	0
Fact_Performance	4	20250101	10
Fact_Performance	5	20260101	0

Data masih banyak yang null karena masih menggunakan data dummy.

D. MEMBUAT STAGING TABLES



The screenshot shows a SQL Server Management Studio window with a query editor containing several CREATE TABLE statements. The code is organized into sections: 'MENBUAT STAGING TABLES' and 'Buat Schema khusus Staging agar rapi'. The first section creates a schema 'Staging' and four staging tables: 'Staging_MsPegawai' (with columns NIP, Namalengkap, TglLahir, JenisKelamin, StatusPegawai, KodeUnit, KodeJabatan, KodeGolongan), 'Staging_TraAbsensi' (with columns NIP, Tanggal, JamMasuk, JamKeluar, StatusKehadiran), 'Staging_TraGaji' (with columns NIP, Bulan, Tahun, GajiPokok), and 'Staging_TraPenilaian' (with columns NIP, TglEvaluasi, TotalSkor, Grade). The second section is a comment block. The code uses GO statements to separate the schema creation from the table definitions.

```
-- MENBUAT STAGING TABLES
-- =====
-- Buat Schema khusus Staging agar rapi
CREATE SCHEMA Staging;
GO

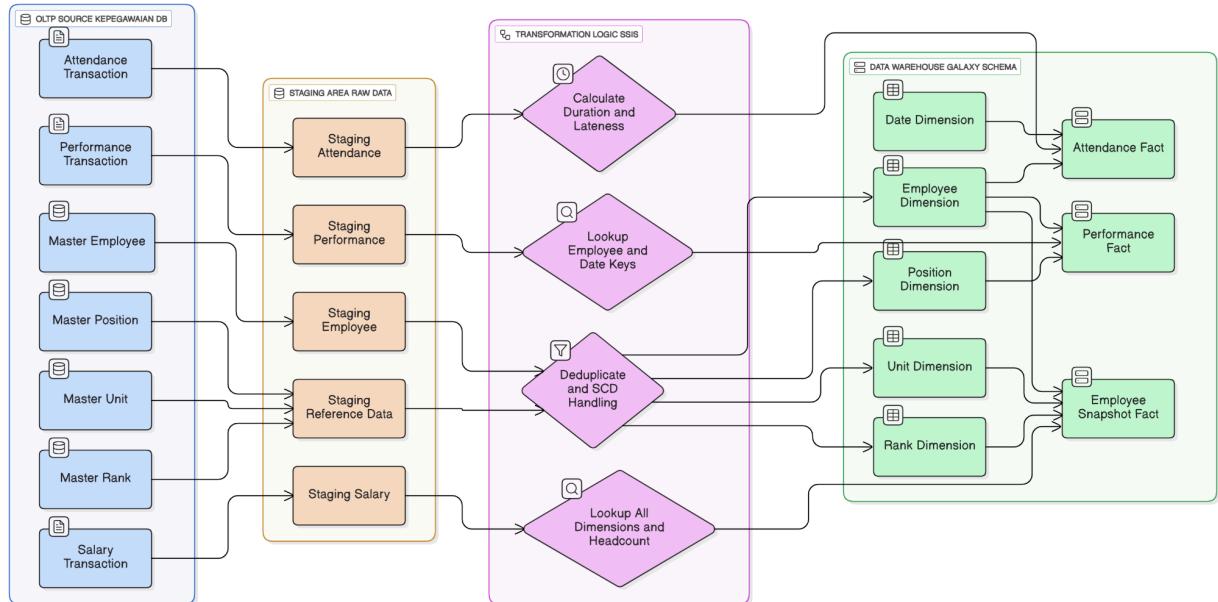
-- Buat tabel Staging (Strukturnya mirip OLTP tapi tidak ada FK/Constraint ketat)
CREATE TABLE Staging_MsPegawai (
    NIP VARCHAR(20),
    Namalengkap VARCHAR(150),
    TglLahir DATE,
    JenisKelamin VARCHAR(10),
    StatusPegawai VARCHAR(50),
    KodeUnit VARCHAR(10),
    KodeJabatan VARCHAR(10),
    KodeGolongan VARCHAR(10)
);

CREATE TABLE Staging_TraAbsensi (
    NIP VARCHAR(20),
    Tanggal DATE,
    JamMasuk TIME(0),
    JamKeluar TIME(0),
    StatusKehadiran VARCHAR(20)
);

CREATE TABLE Staging_TraGaji (
    NIP VARCHAR(20),
    Bulan INT,
    Tahun INT,
    GajiPokok DECIMAL(18,2)
);

CREATE TABLE Staging_TraPenilaian (
    NIP VARCHAR(20),
    TglEvaluasi DATE,
    TotalSkor DECIMAL(5,2),
    Grade VARCHAR(5)
);
GO
```

E. ETL DESIGN



eraser

Strategi ETL menggunakan Script, menggunakan bahasa **T-SQL (Transact-SQL)** yang spesifik untuk lingkungan **Microsoft SQL Server**. Berikut adalah rincian perintah dan logika kunci yang digunakan dalam *script*:

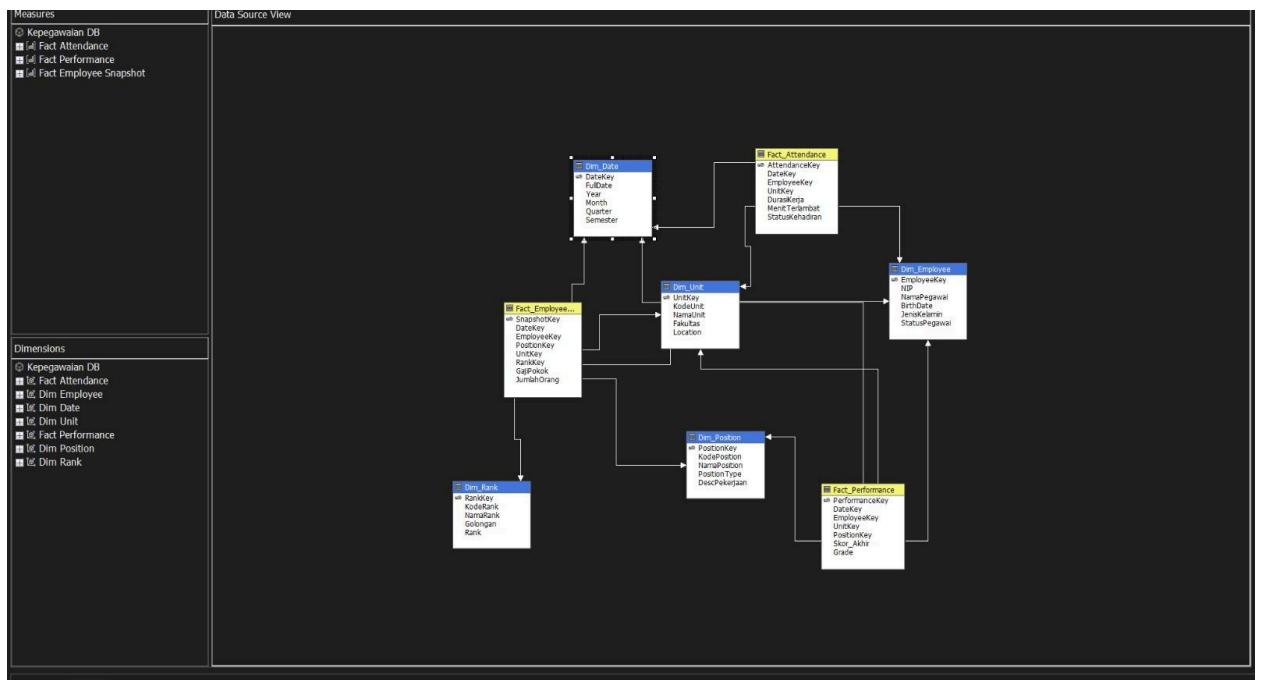
- a. **Control Flow:**
 - i. **CREATE OR ALTER PROCEDURE**: Membungkus seluruh logika dalam satu objek yang mudah dieksekusi ulang atau diperbarui.
 - ii. **SET NOCOUNT ON**: Mengoptimalkan kinerja jaringan dengan mencegah pengiriman pesan "x rows affected" yang tidak perlu.
 - iii. **DECLARE ... GETDATE()**: Digunakan untuk *logging* durasi proses.
- b. **Data Manipulation (DML):**
 - i. **TRUNCATE TABLE**: Membersihkan tabel *Staging* dengan cepat sebelum data baru dimuat (Reset log).
 - ii. **INSERT INTO ... SELECT**: Memindahkan data secara massal (*bulk insert*).
 - iii. **MERGE**: Digunakan pada **Dim_Employee** untuk menangani logika *Upsert* (Update jika ada, Insert jika baru).
 - iv. **DELETE**: Menghapus data spesifik pada Fact Table sebelum insert ulang untuk mencegah duplikasi.
- c. **Transformasi & Fungsi:**
 - i. **DATEDIFF**: Menghitung durasi kerja dan keterlambatan (dalam menit).

- ii. **CAST/CONVERT**: Mengubah format tanggal menjadi **DateKey** (format integer **YYYYMMDD**).
- iii. **ISNULL**: Menangani nilai *NULL* dengan memberikan nilai *default* (biasanya **-1** untuk *Unknown Member*).
- iv. **CASE WHEN**: Logika bisnis untuk menentukan apakah pegawai terlambat (setelah 07:30).

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**.
 - o **Dimensions**: Tabel referensi (**Dim_Unit**, **Dim_Employee**).
 - o **Facts**: Tabel transaksi bisnis yang berisi metrik (**Fact_Attendance**, **Fact_Performance**, **Fact_Employee_Snapshot**).

F. ETL DESIGN TERIMPLEMENTASI



Design ETL terimplementasi menggunakan SSIS Datacube Visual Studio, walaupun ETL yang kami gunakan menggunakan Script untuk Proses ETL nya.

GENERATOR-SCR...\\Fadhil (53)) 08_Data_Quali...\\Fadhil (52)) X 09_Perform

```
1 | EXEC ETL_Master_Load;
```

100 % ✓ No issues found

Messages

```
==== MULAI PROSES ETL ====
1. Loading Staging Tables...
2. Loading Dimensions...
3. Loading Facts...
==== ETL SELESAI ====
Durasi: 1186 ms

Completion time: 2025-11-24T21:19:34.3106918+07:00
```

ETL Berjalan dengan lancar menggunakan Script T-Sql + Staging.

G. DATA LOADED SUCCESSFULLY

a. Dim_Date

DateKey	FullDate	Year	Month	Quarter	Semester
7	2020-01-06	2020	1	1	1
8	2020-01-07	2020	1	1	1
9	2020-01-08	2020	1	1	1
10	2020-01-09	2020	1	1	1
11	2020-01-10	2020	1	1	1
12	2020-01-11	2020	1	1	1
13	2020-01-12	2020	1	1	1
14	2020-01-13	2020	1	1	1
15	2020-01-14	2020	1	1	1
16	2020-01-15	2020	1	1	1
17	2020-01-16	2020	1	1	1
18	2020-01-17	2020	1	1	1
19	2020-01-18	2020	1	1	1
20	2020-01-19	2020	1	1	1
21	2020-01-20	2020	1	1	1
22	2020-01-21	2020	1	1	1
23	2020-01-22	2020	1	1	1
24	2020-01-23	2020	1	1	1
25	2020-01-24	2020	1	1	1
26	2020-01-25	2020	1	1	1
27	2020-01-26	2020	1	1	1
28	2020-01-27	2020	1	1	1
29	2020-01-28	2020	1	1	1
30	2020-01-29	2020	1	1	1
31	2020-01-30	2020	1	1	1
32	2020-01-31	2020	1	1	1
33	2020-02-01	2020	2	1	1
34	2020-02-02	2020	2	1	1
35	2020-02-03	2020	2	1	1
36	2020-02-04	2020	2	1	1
37	2020-02-05	2020	2	1	1
38	2020-02-06	2020	2	1	1
39	2020-02-07	2020	2	1	1
40	2020-02-08	2020	2	1	1
41	2020-02-09	2020	2	1	1
42	2020-02-10	2020	2	1	1
43	2020-02-11	2020	2	1	1
44	2020-02-12	2020	2	1	1
45	2020-02-13	2020	2	1	1

Query executed successfully.

b. Dim_Employee

EmployeeKey	NIP	NamaPegawai	BirthDate	JenisKelamin	StatusPegawai
2	P001	Dr. Budi Santoso	1980-01-01	L	PNS
3	P002	Siti Amriah, M.Si	1990-05-05	P	PNS
4	P003	Rudi Hernawan, S.Kom	1985-03-10	L	PNS
5	P004	Andi Pratama	1995-07-07	L	Kontrak
6	P005	Ir. Joko Susilo	1975-09-09	L	PNS
7	P006	Rina Watt, A.Md	1998-01-01	P	Kontrak
8	P007	Dewi Lestari, M.Sc	1992-12-12	P	Tetap Non-PNS
9	P008	Fajar Nugraha, Ph.D	1988-08-08	L	PNS
10	P009	Maya Sari	1996-04-04	P	Kontrak
11	P010	Eko Prasetyo	1993-11-11	L	Tetap Non-PNS
12	AUTO-00001	Dinda Reza Nasution	1986-11-24	L	Tetap
13	AUTO-00002	Hendra Bunga Pratama	1979-11-24	P	Tetap
14	AUTO-00003	Bagas Selia Utama	1999-11-24	L	Tetap
15	AUTO-00004	Ricky Hadi Wijaya	1986-11-24	L	Tetap
16	AUTO-00005	Zanal Feri Hidayat	1999-11-24	P	Tetap
17	AUTO-00006	Oki Dian Sriogear	1989-11-24	P	Tetap
18	AUTO-00007	Agus Ajy Kusuma	2001-11-24	P	Tetap
19	AUTO-00008	Agus Akbar Pohan	1994-11-24	L	Tetap
20	AUTO-00009	Dinda Ayu Santoso	1977-11-24	P	Tetap
21	AUTO-00010	Zanal Tri Saputra	2000-11-24	L	Tetap
22	AUTO-00011	Siti Dian Ramadhan	1995-11-24	P	Tetap
23	AUTO-00012	Vina Sari Selawati	1983-11-24	P	Tetap
24	AUTO-00013	Citra Ajy Shombing	1987-11-24	P	Tetap
25	AUTO-00014	Nur Bayu Ramadhan	1985-11-24	P	Tetap
26	AUTO-00015	Muhammad Bagus Permana	1996-11-24	L	Tetap
27	AUTO-00016	Yusuf Kiki Utama	1999-11-24	P	Tetap
28	AUTO-00017	Indah Sari Susanto	1978-11-24	L	Tetap
29	AUTO-00018	Tio Bagus Permana	1974-11-24	L	Tetap
30	AUTO-00019	Oki Akbar Saputra	1989-11-24	L	Tetap
31	AUTO-00020	Ratna Dian Wijaya	1986-11-24	L	Tetap
32	AUTO-00021	Indah Nur Permania	1981-11-24	P	Tetap
33	AUTO-00022	Vina Jaya Kusuma	1981-11-24	P	Tetap
34	AUTO-00023	Joko Sari Hidayat	1977-11-24	P	Tetap
35	AUTO-00024	Ahmad Feri Wibowo	1997-11-24	P	Tetap
36	AUTO-00025	Ratna Kusuma Lesmana	1984-11-24	L	Tetap
37	AUTO-00026	Nur Lu Pohan	1990-11-24	L	Tetap
38	AUTO-00027	Nur Kusuma Siregar	1992-11-24	L	Tetap
39	AUTO-00028	Oki Bunga Pohan	1997-11-24	P	Tetap

Query executed successfully.

c. Dim_Position

SQLQuery2.sq...Fadhil (69)* 05_Partition_I...EE\Fadhil (72)					
1	select * from Dim_Position;				
100 %	✖ 1	▲ 0	↑	↓	
Results Messages					
	PositionKey	KodePosition	NamaPosition	PositionType	DescPekerjaan
1	1	AA	Asisten Ahli	Fungsional	Pengajar Junior
2	2	KBG	Kepala Bagian	Struktural	Pimpinan Unit
3	3	LBR	Laboran	Fungsional	Teknis Lab
4	4	LEC	Dosen Lektor	Fungsional	Pengajar Senior
5	5	STF	Staff Administrasi	Struktural	Administrasi

d. Dim_Rank

SQLQuery2.sq...Fadhil (69)* 05_Partition_I...EE\Fadhil (72))					
1	Select * from Dim_Rank				
100 %	✖ 1	▲ 0	↑	↓	
Results Messages					
	RankKey	KodeRank	NamaRank	Golongan	Rank
1	1	III-A	Penata Muda	III/a	NULL
2	2	III-B	Penata Muda Tk. I	III/b	NULL
3	3	III-C	Penata	III/c	NULL
4	4	IV-A	Pembina	IV/a	NULL
5	5	IV-B	Pembina Tk. I	IV/b	NULL

e. Dim_Unit

SQLQuery2.sq...Fadhil (69)* 05_Partition_I...EE\Fadhil (72))					
1	select * from Dim_Unit				
100 %	✖	No issues found	◀	▶	
Results Messages					
	UnitKey	KodeUnit	NamaUnit	Fakultas	Location
1	1	BIO	Biologi	Fakultas Sains	Gedung E
2	2	HUM	Biro SDM & Umum	Rektorat	Gedung A
3	3	IF	Teknik Informatika	Fakultas Teknologi	Gedung F
4	4	SD	Sains Data	Fakultas Sains	Gedung E
5	5	TIP	Teknik Industri Pertanian	Fakultas Teknologi	Gedung F

H. DATA QUALITY CHECK

- a. Apakah jumlah baris di Source sama dengan Target?
- b. Apakah ada EmployeeKey yang -1 (Unknown)? Jika ada, berarti NIP di transaksi tidak ada di Master Pegawai.
- c. Validasi Perhitungan Measure Pastikan DurasiKerja masuk akal.

```
-- Cek 1: Apakah jumlah baris di Source sama dengan Target?  
SELECT  
    (SELECT COUNT(*) FROM TrAbsensi) AS Source_Absensi,  
    (SELECT COUNT(*) FROM Fact_Attendance) AS Target_Absensi;  
  
-- Cek 2: Apakah ada EmployeeKey yang -1 (Unknown)?  
-- Jika ada, berarti NIP di transaksi tidak ada di Master Pegawai  
SELECT * FROM Fact_Attendance WHERE EmployeeKey = -1;  
  
-- Cek 3: Validasi Perhitungan Measure  
-- Pastikan DurasiKerja masuk akal  
SELECT TOP 5 * FROM Fact_Attendance ORDER BY DurasiKerja DESC;
```

	Source_Absensi	Target_Absensi
1	20000	20000

	AttendanceKey	DateKey	EmployeeKey	UnitKey	DurasiKerja	MenitTerlambat	StatusKehadiran
1	14	20251119	18	5	9.00	0	Hadir
2	22	20250917	18	5	9.00	0	Hadir
3	30	20250826	18	5	9.00	0	Hadir
4	35	20250801	18	5	9.00	0	Hadir
5	69	20250926	18	5	9.00	0	Hadir

I. PERFORMANCE TESTING

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top window is titled "SQLQuery2.sq...Fadhil (69)*" and "05_Partition_I...EE\Fadhil (72)". It contains a T-SQL script for performing a performance test on an ETL process. The script uses `PRINT` statements to log start and end times, and calculates the duration in milliseconds. The bottom window is titled "Messages" and displays the execution results, including the start time ("--- START PERFORMANCE TEST: ETL PROCESS ---"), steps of the ETL process ("1. Loading Staging Tables...", "2. Loading Dimensions...", "3. Loading Facts..."), completion time ("--- ETL SELESAI ---"), duration ("Durasi: 1250 ms"), and end time ("Waktu Mulai : 2025-11-24 21:05:34.230", "Waktu Selesai : 2025-11-24 21:05:35.600", "Total Durasi : 1370 milidetik").

```
1 USE Kepergawaiian_DB;
2 GO
3
4 PRINT '--- START PERFORMANCE TEST: ETL PROCESS ---';
5 DECLARE @StartTime DATETIME = GETDATE();
6
7 -- Jalankan Proses ETL Utama
8 EXEC ETL_Master_Load;
9
10 DECLARE @EndTime DATETIME = GETDATE();
11 DECLARE @DurationMs INT = DATEDIFF(MILLISECOND, @StartTime, @EndTime);
12
13 PRINT '-----';
14 PRINT 'Waktu Mulai : ' + CONVERT(VARCHAR, @StartTime, 121);
15 PRINT 'Waktu Selesai : ' + CONVERT(VARCHAR, @EndTime, 121);
16 PRINT 'Total Durasi : ' + CAST(@DurationMs AS VARCHAR) + ' milidetik';
17 PRINT '-----';
```

100 % ✖ 1 ⚠ 0 ↑ ↓

Messages

```
--- START PERFORMANCE TEST: ETL PROCESS ---
==== MULAI PROSES ETL ====
1. Loading Staging Tables...
2. Loading Dimensions...
3. Loading Facts...
==== ETL SELESAI ====
Durasi: 1250 ms
-----
Waktu Mulai : 2025-11-24 21:05:34.230
Waktu Selesai : 2025-11-24 21:05:35.600
Total Durasi : 1370 milidetik
-----
Completion time: 2025-11-24T21:05:35.6308128+07:00
```