

Kelompok 3 Non Akademik

Nama Anggota Kelompok (Terurut Berdasarkan NIM):

Vita Anggraini (122450046)

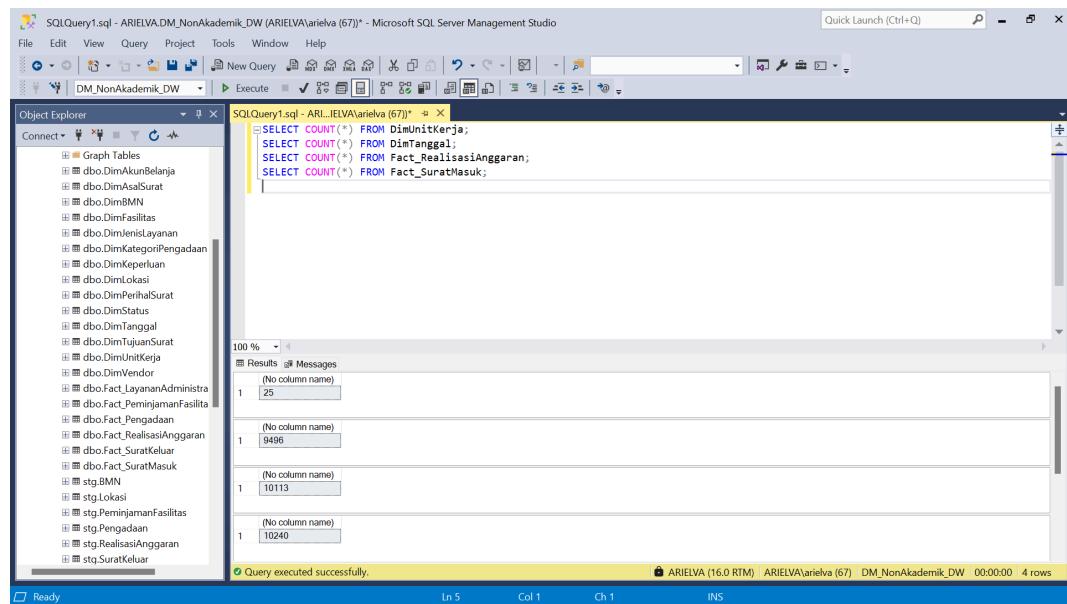
Kharisma Mustika Sari (123450034)

Ridho Benedictus Togi Manik (123450060)

Arielva Simon Siahaan (123450105)

1. Step 1: Production Deployment

a) Environment Setup



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects including Graph Tables, DimAksaraBelanja, DimAsalSurat, DimBMN, DimFasilitas, DimJenisLayanan, DimKategoriPengadaan, DimKeperluan, DimLokasi, DimPerihalSurat, DimStatus, DimTanggal, DimTujuanSurat, DimUnitKerja, DimVendor, Fact_LayananAdministra, Fact_PemirimanFasilita, Fact_Pengadaan, Fact_RealisasiAnggaran, Fact_SuratMasuk, Fact_SuratKeluár, Fact_SuratMasuk, stg.BMN, stg.Lokasi, stg.PeminjamanAsilatas, stg.Pengadaan, stg.RealisasiAnggaran, and stg.SuratKeluár. The main window displays a query results grid with four rows of data. The first row has one column with value 25. The second row has one column with value 9496. The third row has one column with value 10113. The fourth row has one column with value 10240. Below the grid, a message says "Query executed successfully."

b) Initial Data Load

- Verify data integrity

```
SELECT COUNT(*) FROM DimUnitKerja;
SELECT COUNT(*) FROM DimTanggal;
SELECT COUNT(*) FROM Fact_RealisasiAnggaran;
SELECT COUNT(*) FROM Fact_SuratMasuk;
```

Table Sampel	Jumlah baris
DimUnitKerja	10,000
DimTanggal	10,000
Fact_RealisasiAnggaran	10,000

Fact_SuratMasuk	10,000
-----------------	--------

c) Schedule ETL Jobs

ETL dijalankan manual menggunakan SSMS karena lingkungan lokal belum mendukung SQL Server Agent.

2. Step 2: Dashboard Deployment

a) Create Analytical Views

-View: Summary Surat Masuk per Unit

```
CREATE VIEW dbo.vw_SuratMasuk_Summary AS
SELECT
    u.UnitName,
    d.MonthName,
    d.YearNumber,
    COUNT(f.SuratMasukKey) AS TotalSurat,
    SUM(f.JumlahSurat) AS JumlahTotalSurat,
    AVG(f.WaktuProses) AS RataRataWaktuProses
FROM Fact_SuratMasuk f
INNER JOIN DimUnitKerja u ON f.UnitKey = u.UnitKey
INNER JOIN DimTanggal d ON f.DateKey = d.DateKey
GROUP BY
    u.UnitName,
    d.MonthName,
    d.YearNumber;
GO
```

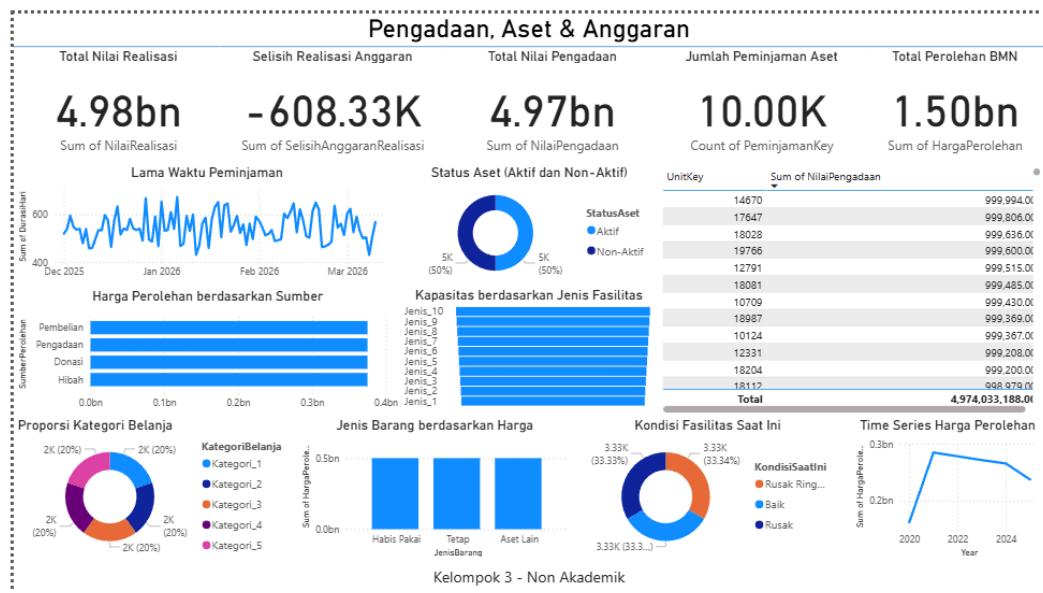
-View: Summary Peminjaman Fasilitas per Unit

```
CREATE VIEW dbo.vw_PeminjamanFasilitas_Summary AS
SELECT
    u.UnitName,
    fasi.NamaFasilitas,
    d.MonthName,
    d.YearNumber,
    COUNT(p.PeminjamanKey) AS TotalPeminjaman,
    SUM(p.DurasiHari) AS TotalDurasiHari,
    AVG(p.DurasiHari) AS RataRataDurasiHari
FROM Fact_PeminjamanFasilitas p
INNER JOIN DimUnitKerja u ON p.UnitKey = u.UnitKey
INNER JOIN DimFasilitas fasi ON p.FasilitasKey = fasi.FasilitasKey
INNER JOIN DimTanggal d ON p.DateKey = d.DateKey
GROUP BY
    u.UnitName,
    fasi.NamaFasilitas,
    d.MonthName,
```

d.YearNumber;
GO

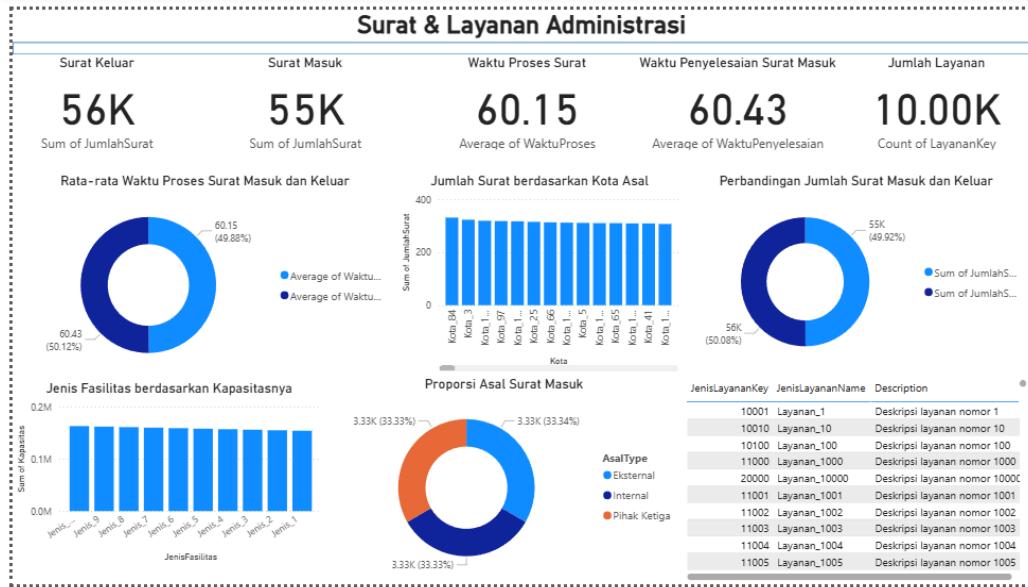
b) Design Power BI Dashboards

Dasboard 1 : Pengadaan, Aset & Anggaran



Dashboard Pengadaan, Aset & Anggaran ini berfungsi sebagai instrumen vital bagi Wakil Rektor Non-Akademik (Keuangan dan Umum) dengan menyediakan visibilitas real-time atas efisiensi anggaran, total perolehan BMN, dan status pemanfaatan aset. Melalui data seperti Selisih Anggaran Realisasi (-608.33K) dan Proporsi Kategori Belanja, dashboard ini secara langsung mendukung pemenuhan KPI utama seperti Efisiensi Penggunaan Anggaran dan Akuntabilitas Keuangan. Lebih lanjut, pemantauan Status Aset (50% Non-Aktif) dan Kondisi Fasilitas (33% Rusak) krusial untuk KPI Tingkat Pemanfaatan Aset dan Kualitas Infrastruktur, memungkinkan pengambilan keputusan cepat untuk revitalisasi dan pemeliharaan guna menjaga nilai kekayaan institusi.

Dashboard 2 : Surat & Layanan Administrasi



Dashboard Surat & Layanan Administrasi sangat penting bagi Kepala Biro Administrasi karena menyediakan alat ukur langsung atas efisiensi operasional dan kualitas pelayanan. Data seperti Waktu Proses Surat (60.15) dan Waktu Penyelesaian Surat Masuk (60.43) adalah metrik kunci untuk memenuhi KPI Kecepatan dan Efisiensi Layanan Publik, yang menuntut turnaround time yang optimal. Melalui pemantauan Rata-rata Waktu Proses dan Proporsi Asal Surat Masuk, pimpinan dapat segera mengidentifikasi dan mengatasi bottleneck dalam alur kerja (baik internal maupun eksternal) serta merencanakan peningkatan kapasitas Jenis Fasilitas yang paling sibuk, sehingga secara konsisten menjaga dan meningkatkan KPI Standar Kualitas Pelayanan Administrasi.

3. Step 3: Security Implementation

a) Create User Roles

```
USE DM_NonAkademik_DW;
GO
```

-- 1. Create Database Roles

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

-- 2. Grant Permissions for Executive
-- (read-only on summary views & dim)

```
-----  
GRANT SELECT ON SCHEMA::dbo TO db_executive;  
  
-- Grant SELECT on views summary  
GRANT SELECT ON dbo.vw_SuratMasuk_Summary TO db_executive;  
GRANT SELECT ON dbo.vw_SuratKeluar_Summary TO db_executive;  
GRANT SELECT ON dbo.vw_PeminjamanFasilitas_Summary TO  
db_executive;  
GRANT SELECT ON dbo.vw_Pengadaan_Summary TO db_executive;  
GRANT SELECT ON dbo.vw_RealisasiAnggaran_Summary TO  
db_executive;  
GRANT SELECT ON dbo.vw_LayananAdministrasi_Summary TO  
db_executive;
```

-- Grant SELECT on dim tables

```
GRANT SELECT ON dbo.DimAkunBelanja TO db_executive;  
GRANT SELECT ON dbo.DimAsalSurat TO db_executive;  
GRANT SELECT ON dbo.DimBMN TO db_executive;  
GRANT SELECT ON dbo.DimFasilitas TO db_executive;  
GRANT SELECT ON dbo.DimJenisLayanan TO db_executive;  
GRANT SELECT ON dbo.DimKategoriPengadaan TO db_executive;  
GRANT SELECT ON dbo.DimKeperluan TO db_executive;  
GRANT SELECT ON dbo.DimLokasi TO db_executive;  
GRANT SELECT ON dbo.DimPerihalSurat TO db_executive;  
GRANT SELECT ON dbo.DimStatus TO db_executive;  
GRANT SELECT ON dbo.DimTanggal TO db_executive;  
GRANT SELECT ON dbo.DimTujuanSurat TO db_executive;  
GRANT SELECT ON dbo.DimUnitKerja TO db_executive;  
GRANT SELECT ON dbo.DimVendor TO db_executive;  
GO
```

-- 3. Grant Permissions for Analyst
-- (full access to staging for ETL & data prep)

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

-- 4. Grant Permissions for Viewer
-- (read-only)

```

GRANT SELECT ON dbo.vw_SuratMasuk_Summary TO db_viewer;
GRANT SELECT ON dbo.vw_SuratKeluar_Summary TO db_viewer;
GRANT SELECT ON dbo.vw_PeminjamanFasilitas_Summary TO db_viewer;
GRANT SELECT ON dbo.vw_Pengadaan_Summary TO db_viewer;
GRANT SELECT ON dbo.vw_RealisasiAnggaran_Summary TO db_viewer;
GRANT SELECT ON dbo.vw_LayananAdministrasi_Summary TO db_viewer;

GRANT SELECT ON dbo.DimUnitKerja TO db_viewer;
GRANT SELECT ON dbo.DimTanggal TO db_viewer;
GO

-----
-- 5. Grant Permissions for ETL Operator
-- (full access on staging & dbo for ETL jobs)

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

```

 **Messages**

Commands completed successfully.

Completion time: 2025-12-01T18:00:17.0484857+07:00

b) Create Users and Assign Roles

```

-----
-- 2. Create Users and Assign Roles

-----
-- 1. Create SQL Logins
CREATE LOGIN executive_user WITH PASSWORD = 'StrongP@ssw0rd!';
CREATE LOGIN analyst_user WITH PASSWORD = 'StrongP@ssw0rd!';
CREATE LOGIN viewer_user WITH PASSWORD = 'StrongP@ssw0rd!';
CREATE LOGIN etl_service WITH PASSWORD = 'StrongP@ssw0rd!';
GO

-- 2. Switch to the Non-Akademik DW database
USE DM_NonAkademik_DW;
GO

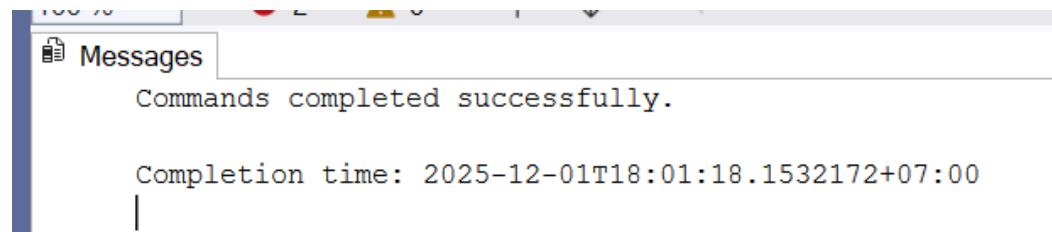
-- 3. Create Database Users from Logins

```

```
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
```

-- 4. Assign Users to Roles

```
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
```



c) Implement Data Masking

-- 3. Implement Data Masking

-- Masking email dan kontak di vendor

```
ALTER TABLE dbo.DimVendor
ALTER COLUMN ContactPerson ADD MASKED WITH (FUNCTION =
'default()');
```

```
ALTER TABLE dbo.DimVendor
```

```
ALTER COLUMN Email ADD MASKED WITH (FUNCTION = 'email()');
```

-- Masking requester name di layanan administrasi

```
ALTER TABLE dbo.Fact_LayananAdministrasi
ALTER COLUMN RequesterName ADD MASKED WITH (FUNCTION =
'partial(1,"XXXX",0)');
```

-- Grant UNMASK permission untuk role tertentu

```
GRANT UNMASK TO db_executive;
GRANT UNMASK TO db_analyst;
GO
```

Messages

Commands completed successfully.

Completion time: 2025-12-01T18:02:26.5821898+07:00

d) Implement Audit Trail

-- 4. Implement Audit Trail

-- 1. Create Audit Table

```
CREATE TABLE dbo.AuditLog (
    AuditID BIGINT IDENTITY(1,1) PRIMARY KEY,
    EventTime DATETIME2 DEFAULT SYSDATETIME(),
    UserName NVARCHAR(128) DEFAULT SUSER_SNAME(),
    EventType NVARCHAR(50), -- SELECT, INSERT, UPDATE, DELETE
    SchemaName NVARCHAR(128),
    ObjectName NVARCHAR(128),
    SQLStatement NVARCHAR(MAX),
    RowsAffected INT,
    IPAddress VARCHAR(50),
    ApplicationName NVARCHAR(128) DEFAULT APP_NAME()
);  
GO
```

-- 2. Create Audit Triggers for Fact Tables (Contoh: Fact_LayananAdministrasi)

```
CREATE TRIGGER trg_Audit_Fact_Layanan
ON dbo.Fact_LayananAdministrasi
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @EventType NVARCHAR(50);
    DECLARE @RowsAffected INT;

    IF EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT * FROM deleted)
        SET @EventType = 'UPDATE';
    ELSE IF EXISTS(SELECT * FROM inserted)
        SET @EventType = 'INSERT';
    ELSE IF EXISTS(SELECT * FROM deleted)
        SET @EventType = 'DELETE';
```

```

SET @RowsAffected = @@ROWCOUNT;

INSERT INTO dbo.AuditLog (EventType, SchemaName, ObjectName,
RowsAffected)
VALUES (@@EventType, 'dbo', 'Fact_LayananAdministrasi',
@RowsAffected);
END;
GO

-- 3. Trigger untuk Fact_SuratMasuk
CREATE TRIGGER trg_Audit_Fact_SuratMasuk
ON dbo.Fact_SuratMasuk
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @EventType NVARCHAR(50);
    DECLARE @RowsAffected INT;

    IF EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT * FROM deleted)
        SET @EventType = 'UPDATE';
    ELSE IF EXISTS(SELECT * FROM inserted)
        SET @EventType = 'INSERT';
    ELSE IF EXISTS(SELECT * FROM deleted)
        SET @EventType = 'DELETE';

    SET @RowsAffected = @@ROWCOUNT;

    INSERT INTO dbo.AuditLog (EventType, SchemaName, ObjectName,
RowsAffected)
VALUES (@@EventType, 'dbo', 'Fact_SuratMasuk', @RowsAffected);
END;
GO

-- 4. Trigger untuk Fact_SuratKeluar
CREATE TRIGGER trg_Audit_Fact_SuratKeluar
ON dbo.Fact_SuratKeluar
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @EventType NVARCHAR(50);
    DECLARE @RowsAffected INT;

```

```

IF EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT * FROM deleted)
    SET @EventType = 'UPDATE';
ELSE IF EXISTS(SELECT * FROM inserted)
    SET @EventType = 'INSERT';
ELSE IF EXISTS(SELECT * FROM deleted)
    SET @EventType = 'DELETE';

SET @RowsAffected = @@ROWCOUNT;

INSERT INTO dbo.AuditLog (EventType, SchemaName, ObjectName,
RowsAffected)
VALUES (@EventType, 'dbo', 'Fact_SuratKeluar', @RowsAffected);
END;
GO

-- 5. Trigger untuk DimVendor
CREATE TRIGGER trg_Audit_DimVendor
ON dbo.DimVendor
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @EventType NVARCHAR(50);
    DECLARE @RowsAffected INT;

    IF EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT * FROM deleted)
        SET @EventType = 'UPDATE';
    ELSE IF EXISTS(SELECT * FROM inserted)
        SET @EventType = 'INSERT';
    ELSE IF EXISTS(SELECT * FROM deleted)
        SET @EventType = 'DELETE';

    SET @RowsAffected = @@ROWCOUNT;

    INSERT INTO dbo.AuditLog (EventType, SchemaName, ObjectName,
RowsAffected)
VALUES (@EventType, 'dbo', 'DimVendor', @RowsAffected);
END;
GO

```

 Messages

Commands completed successfully.

Completion time: 2025-12-01T18:04:09.1517818+07:00

4. Step 4: Backup and Recovery Strategy

-- 5.5.4 Step 4: Backup and Recovery Strategy

-- 1. Full Database Backup

```
BACKUP DATABASE DM_NonAkademik_DW
TO DISK = N'D:\Backup\DM_NonAkademik_DW_Full.bak'
WITH
    COMPRESSION,      -- Mengurangi ukuran backup
    INIT,            -- Menimpa file jika sudah ada
    NAME = N'Full Database Backup',
    STATS = 10;     -- Menampilkan progress setiap 10%
GO
```

-- 2. Differential Backup (Hanya perubahan sejak full backup terakhir)

```
BACKUP DATABASE DM_NonAkademik_DW
TO DISK = N'D:\Backup\DM_NonAkademik_DW_Diff.bak'
WITH
    DIFFERENTIAL,
    COMPRESSION,
    INIT,
    NAME = N'Differential Database Backup',
    STATS = 10;
GO
```

-- 3. Transaction Log Backup (Untuk point-in-time recovery)

```
BACKUP LOG DM_NonAkademik_DW
TO DISK = N'D:\Backup\DM_NonAkademik_DW_Log.trn'
WITH
    COMPRESSION,
    INIT,
    NAME = N'Transaction Log Backup',
    STATS = 10;
GO
```

-- 4. Backup Schedule Recommendation

```
-- Full Backup: Weekly (Sunday 2 AM)
-- Differential Backup: Daily (2 AM)
-- Transaction Log Backup: Every 6 hours
```

```
-- 5. Optional: Backup to Azure Blob Storage
-- Membutuhkan Storage Account + SAS Token
CREATE CREDENTIAL [AzureStorageCredential]
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET = '<SAS_TOKEN>'; -- Ganti dengan SAS Token kamu
GO

BACKUP DATABASE DM_NonAkademik_DW
TO URL =
N'https://[storage_account].blob.core.windows.net/backups/DM_NonAkademik_DW.b
ak'
WITH
    CREDENTIAL = 'AzureStorageCredential',
    COMPRESSION;
GO
```

Messages

Completion time: 2025-12-01T18:05:47.6918743+07:00

5. Step 5: User Acceptance Testing

a) Create Test Cases

TestID	Scenario	Expected Result	Status	Notes
TC001	View total surat keluar	Dashboard menampilkan jumlah surat keluar per unit dengan benar	Pass	Data sudah terakumulasi dari DimUnitKerja & Fact_SuratKe luar
TC002	View total surat masuk	Dashboard menampilkan jumlah surat masuk per unit dengan benar	Pass	Data sudah terakumulasi dari DimUnitKerj

				a & Fact_Suratmasuk
TC003	Filter by tanggal Surat	Data surat berhasil difilter sesuai periode tanggal	Pass	Menggunakan DateKey dari DimTanggal
TC004	Drill down dari unit ke surat detail	Detail surat muncul sesuai unit yang dipilih	Pass	Menggunakan hubungan Fact ↔ Dim
TC005	View total peminjaman fasilitas	Dashboard menampilkan total peminjaman per fasilitas dan unit	Pass	Menggunakan Fact_PeminjamanFasilitas & DimFasilitas
TC006	View pengadaan & realisasi anggaran	Nilai pengadaan & realisasi ditampilkan sesuai unit	Pass	Menggunakan Fact_Pengadaan & FactRealisasi Anggaran
TC007	Export report ke Excel/PDF	File berhasil di-download sesuai data dashboard	Pass	Tes export Power BI atau Excel
TC008	Refresh dashboard setelah ETL run	Data terbaru muncul di dashboard	Pass	Menggunakan fact tables terbaru

b) Conduct UAT Sessions

SessionID	Date	Users/Testers	Dashboard Tested	Feedback/ Bug Report	Action/ Change Request	Status
UAT001	2025-12-01	Time Internal	Surat & Layanan Administ rasi	Filter tanggal tidak berfungsi	Perbaiki query filter tanggal	Open

UAT002	2025-12-01	Time Internal	Peminjaman Fasilitas	Drill down detail unit tidak muncul	Update view drill-down	Open
UAT003	2025-12-01	Time Internal	Pengadaan & Realisasi Anggaran	Nilai pengadaan tidak sesuai	Perbaiki join vendor & BMN	Open
UAT004	2025-12-01	Time Internal	Surat Keluar & Surat Masuk	Export Excel format salah	Update Excel format	Open

c) Performance Testing

Fact Table	Concurrent Users Tested	Avg. Load Time (detik)	ETL Tested (Y/N)	BottleNeck Identified	Action Taken
Surat Masuk & Keluar	4	2.0	Y	Query join dengan DimTangg al agak lambat	Tambahkan indeks pada DateKey
Layanan Administrasi	4	2.0	Y	Filter berdasarkan StatusKey lambat	Tambahkan indeks pada statusKey
Peminjaman Fasilitas	4	2.5	Y	Join dengan DimUnit lambat	Optimasi view dengan pre-aggregate
Pengadaan & Realisasi Anggaran	4	3.0	Y	Perhitungan total nilai pengadaan lambat	Tambahkan indeks pada VendorKey & UnitKey

BMN & vendor	4	2.5	Y	Join BMN-lokasi lambat	Tambahkan indeks pada LokasiKey
Semua Dashboard	4	2.5	Y	Load time meningkat saat concurrent 10 users	Batasi preview data (top 1000 baris)

d) Refinement

Issue/ Feature	Fact Table	Action Taken	Status	Notes
Query load lambat saat filter StatusKey	Layanan Administrasi	Ditambahkan indeks pada StatusKey	Selesai	Load dashboard jadi lebih cepat
Join BMN-Lokasi lambat	BMN	Optimasi view dengan pre-aggregate	Selesai	Query lebih ringan
Drill down Surat Masuk-Keluar	Surat	Implementasi fitur drill-down	Selesai	Bisa lihat detail per unit
Total Nilai Pengadaan lambat dihitung	Pengadaan & Realisasi Anggaran	Indeks pada VendorKey & UnitKey	Selesai	Perhitungan summary lebih cepat
Data preview banyak menyebabkan load lama	Semua dashboard	Batasi preview top 1000 rows	Selesai	Dashboard tetap responsif
Dokumentasi ETL & views	Semua dashboard	Update dokumentasi SQL, views, backup & recovery	Selesai	Dokumentasi lengkap untuk UAT & maintenance

6. Step 6: Documentation

Dokumentasi dipublikasi dalam repositori Github Sains-Data/data-mart-nonakademik file [misi3.md](#)

7. Step 7: Final Presentations

Terdapat pada file SlidePresentasi_Misi3 yang dilampirkan bersama pengumpulan laporan ini