

Data Engineer

Membuat Data Warehouse untuk
kebutuhan salah satu client ID/X
Partners

Fathurrahman Hernanda Khasan



I am an IoT Engineer and Machine Learning enthusiast focused on developing innovative and impactful technology solutions. I have experience in building IoT systems from scratch, starting from planning, prototyping, to implementing solutions in real-world environments.

I also have strong knowledge in the field of Machine Learning and can apply Machine Learning techniques to improve the efficiency of IoT systems and enhance the overall system performance. I have skills in data processing, predictive model creation, algorithm development, and Machine Learning system optimization.

I am interested in new challenges in the field of IoT and Machine Learning and strive to continue learning and growing

Project Overview

Salah satu client dari ID/X Partners yang bergerak di bidang e-commerce memiliki kebutuhan untuk membuat sebuah Data Warehouse yang berasal dari beberapa tabel dari database sumber. Data Warehouse ini nantinya terdiri dari satu tabel Fact dan beberapa tabel Dimension dengan ketentuan Restore Database Staging dan membuat warehouse baru, melakukan job ETL, dan Membuat Store Procedure (SP) untuk menampilkan summary sales order berdasarkan status pengiriman.

Process

1

Melakukan Import/ Restore Database Staging.

2

Membuat sebuah Database bernama DWH_Project, serta membuat Tabel Fact dan Dimension dari tabel yang ada di database Staging.

3

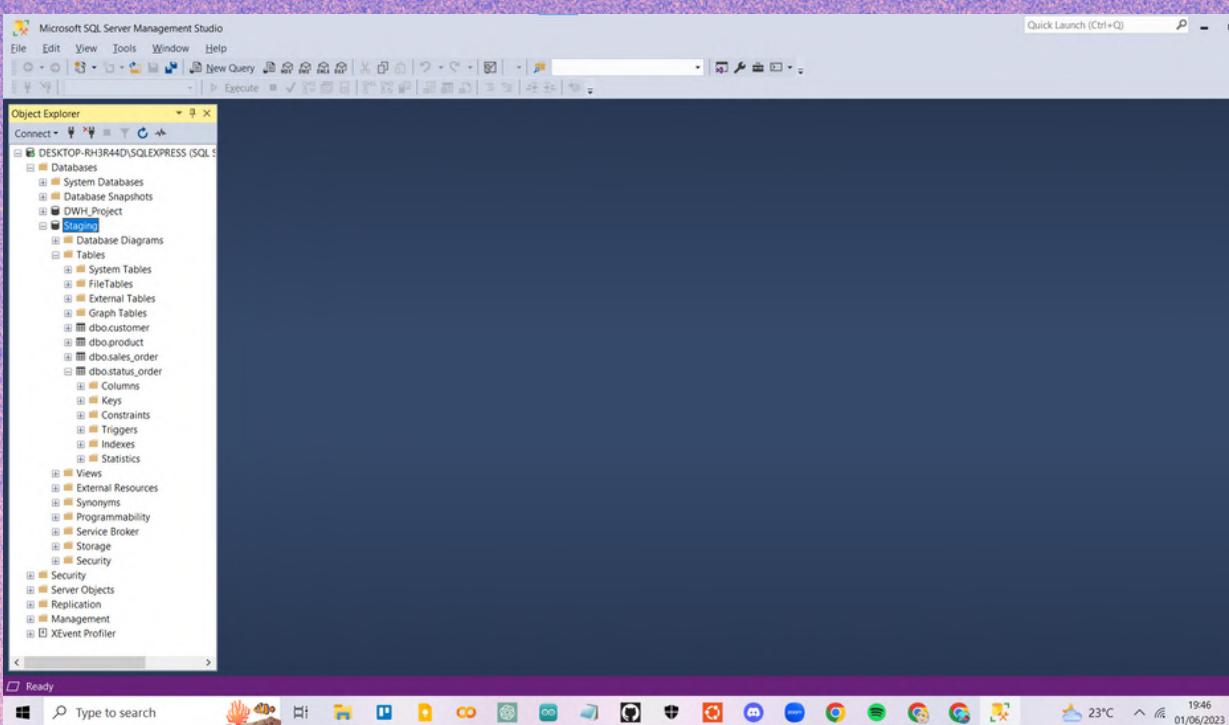
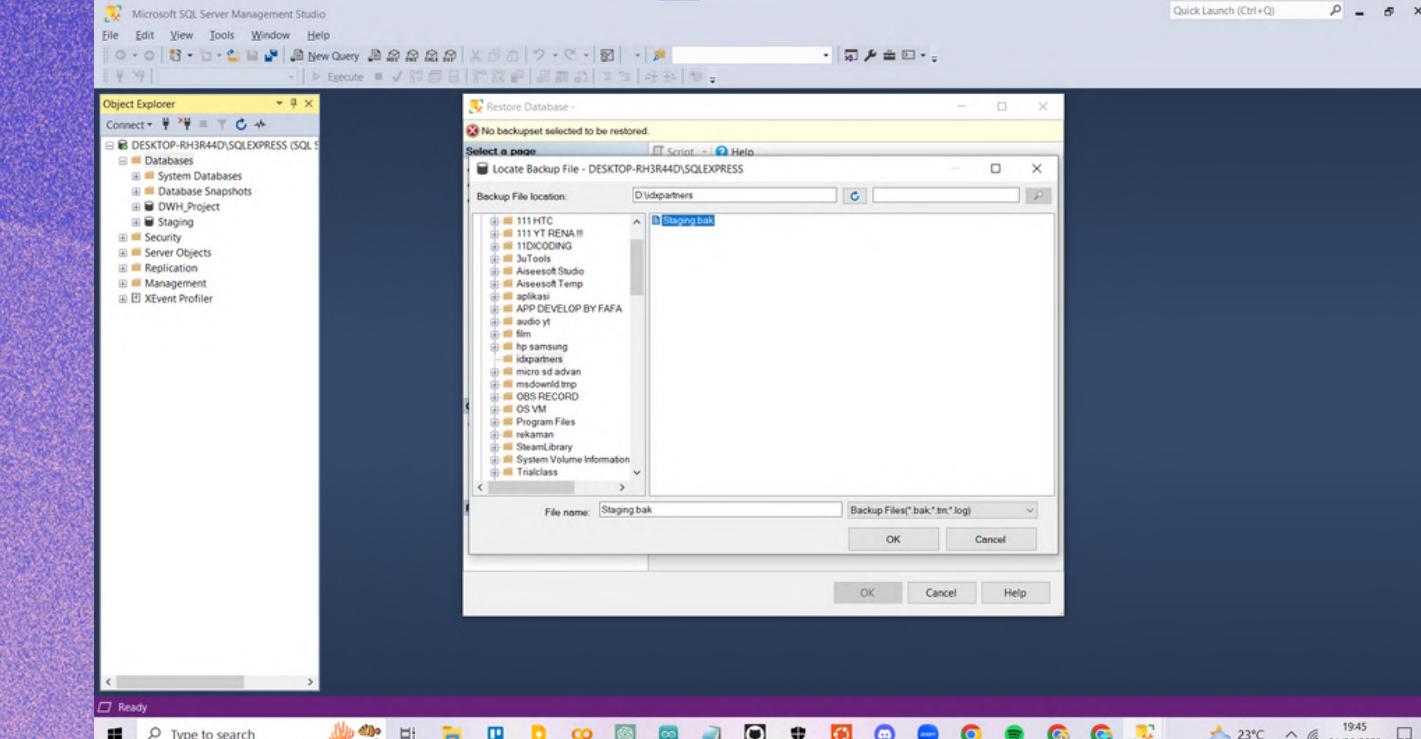
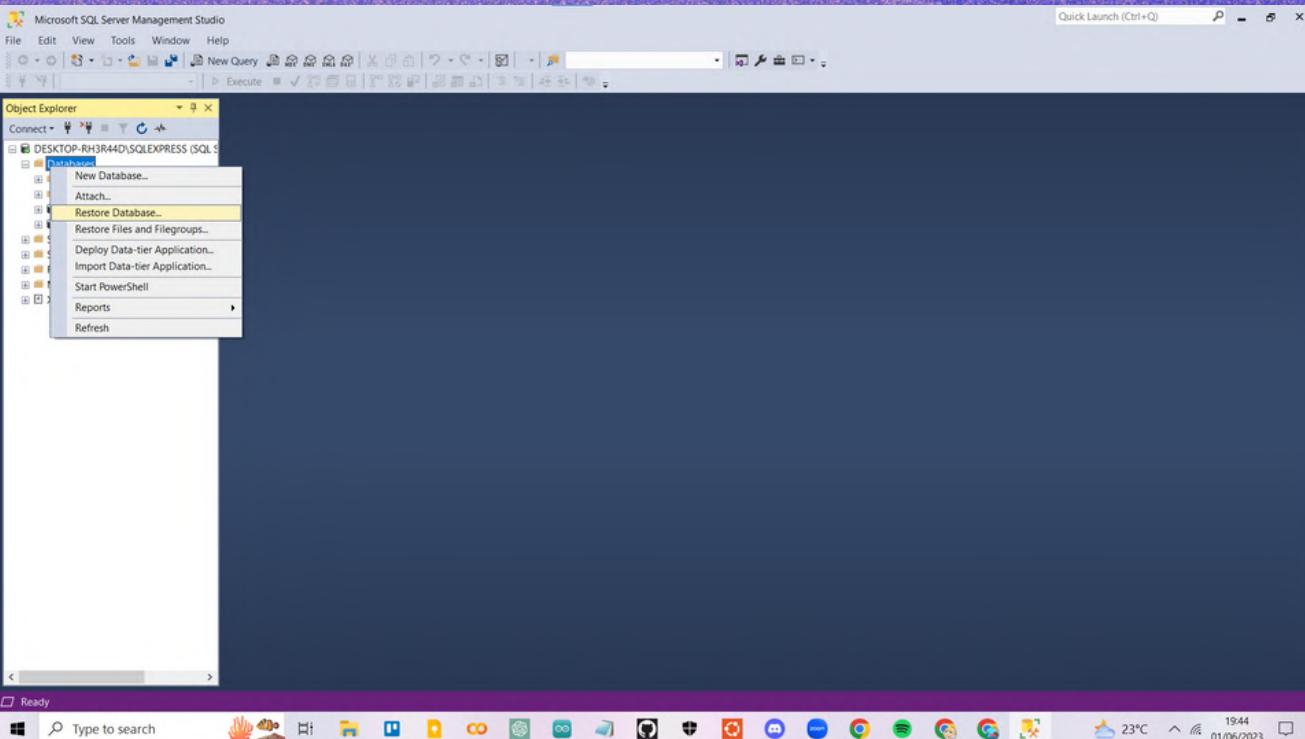
Membuat Job ETL di aplikasi talend untuk memindahkan data dari Staging ke Data Warehouse.

4

Membuat Store Procedure (SP) untuk menampilkan summary sales order berdasarkan status pengiriman.

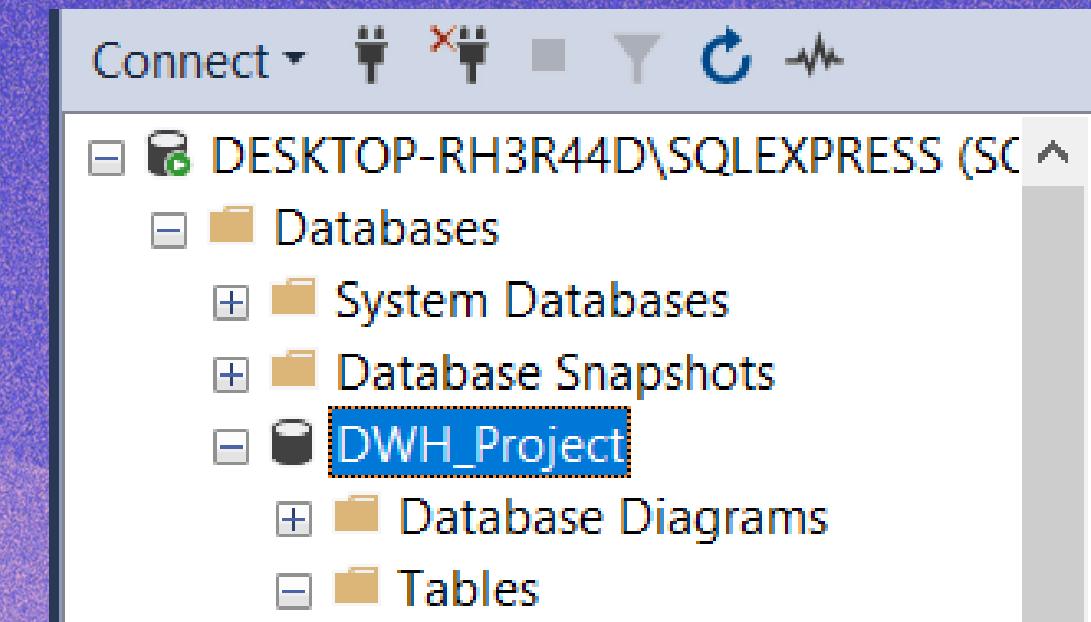
Melakukan Import/ Restore Database Staging.

Melakukan restore database menggunakan file Staging.bak,dengan klik kanan pada database lalu restore database



Membuat sebuah Database bernama DWH_Project, serta membuat Tabel Fact dan Dimension dari tabel yang ada di database Staging

Klik kanan pada Database lalu new Database kemudian, menggunakan perintah CREATE untuk membuat tabel atau manual menggunakan fitur di SSMS untuk membuat tabel yang dibutuhkan dan memberi nama tabel dimension dengan nama DimCustomer, DimProduct, DimStatusOrder dan tabel Fact dengan nama FactSalesOrder



A screenshot showing four separate SSMS Table Designer windows side-by-side. The first window on the left is titled 'Solution1 - DESKTOP-RH3R44D\SQLEXPRESS.DWH_Project - dbo.DimStatusOrder'. It contains columns for StatusID (int), StatusOrder (varchar(50)), and StatusOrderDesc (varchar(50)). The second window is titled 'Solution1 - DESKTOP-RH3R44D\bo.FactSalesOrder'. It contains columns for OrderID (int), CustomerID (int), ProductID (int), Quantity (int), Amount (int), StatusID (int), and OrderDate (date). The third window is titled 'Solution1 - DESKTOP-RH3R44D..dbo.DimCustomer'. It contains columns for CustomerID (int), CustomerName (varchar(50)), Age (int), Gender (varchar(50)), City (varchar(50)), and NoHp (varchar(50)). The fourth window on the right is titled 'Solution1 - DESKTOP-RH3R44D..dbo.DimProduct'. It contains columns for ProductID (int), ProductName (varchar(255)), ProductCategory (varchar(255)), and ProductUnitPrice (int). Each window has its own 'Column Properties' pane at the bottom.

Membuat sebuah Database bernama DWH_Project, serta membuat Tabel Fact dan Dimension dari tabel yang ada di database Staging

Untuk nama-nama kolom dan tipe data dari setiap tabel, ikuti saja dengan yang ada di staging, hanya saja penulisan nama kolom tidak pakai underscore dan setiap kata awalannya pakai huruf kapital (Pascal Case) misal customer_id = CustomerID, product_name = ProductName. Jangan lupa juga memberi primary key dan foreign key pada setiap tabel Fact dan Dimension.

Column Name	Data Type	Allow Nulls
StatusID	int	<input type="checkbox"/>
StatusOrder	varchar(50)	<input type="checkbox"/>
StatusOrderDesc	varchar(50)	<input type="checkbox"/>

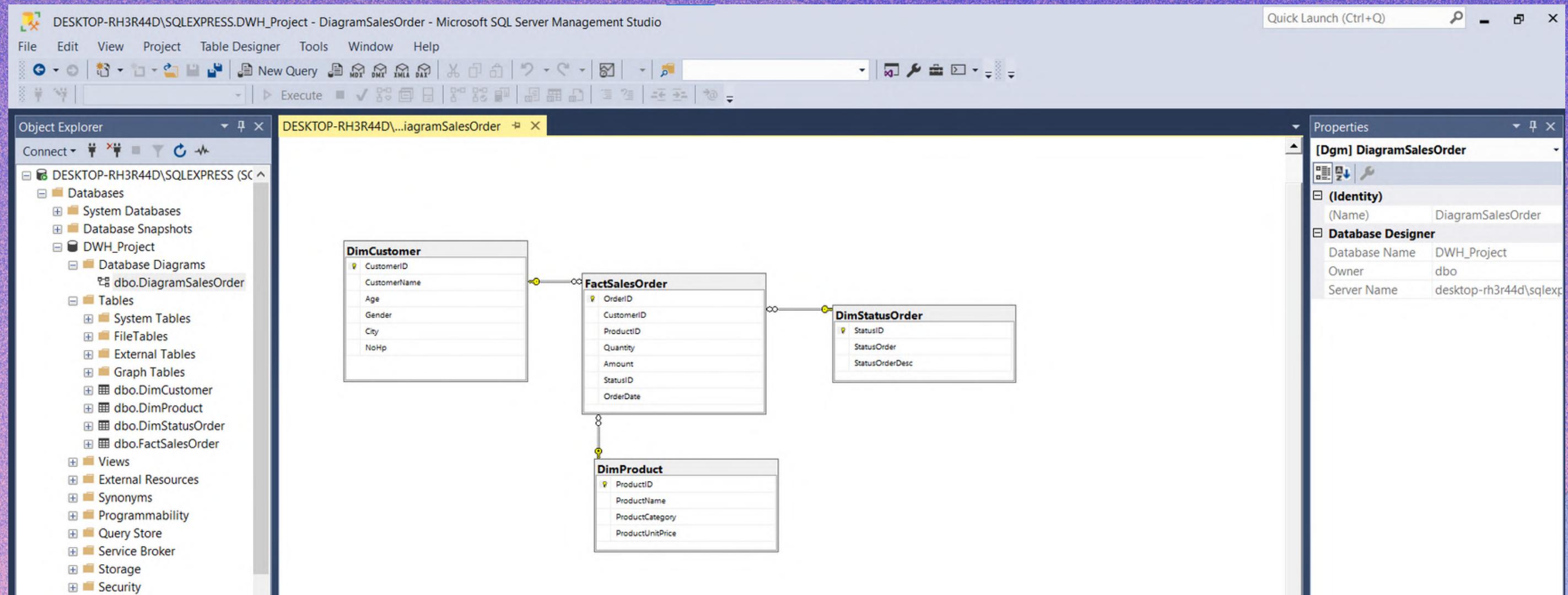
Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
CustomerID	int	<input type="checkbox"/>
ProductID	int	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>
Amount	int	<input type="checkbox"/>
StatusID	int	<input type="checkbox"/>
OrderDate	date	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
ProductName	varchar(255)	<input type="checkbox"/>
ProductCategory	varchar(255)	<input type="checkbox"/>
ProductUnitPrice	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
CustomerName	varchar(50)	<input type="checkbox"/>
Age	int	<input type="checkbox"/>
Gender	varchar(50)	<input type="checkbox"/>
City	varchar(50)	<input type="checkbox"/>
NoHP	varchar(50)	<input type="checkbox"/>

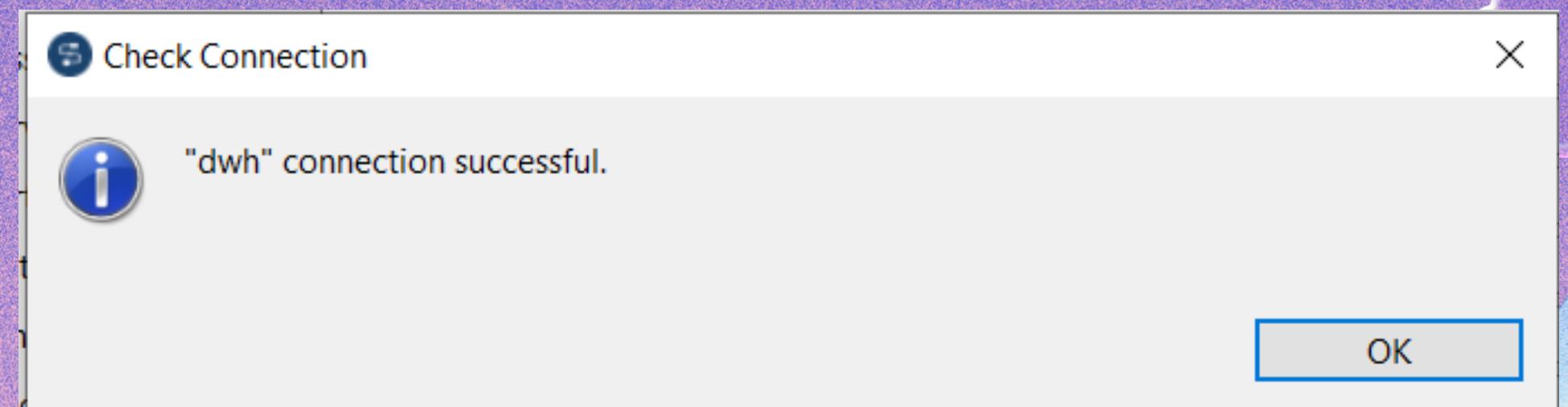
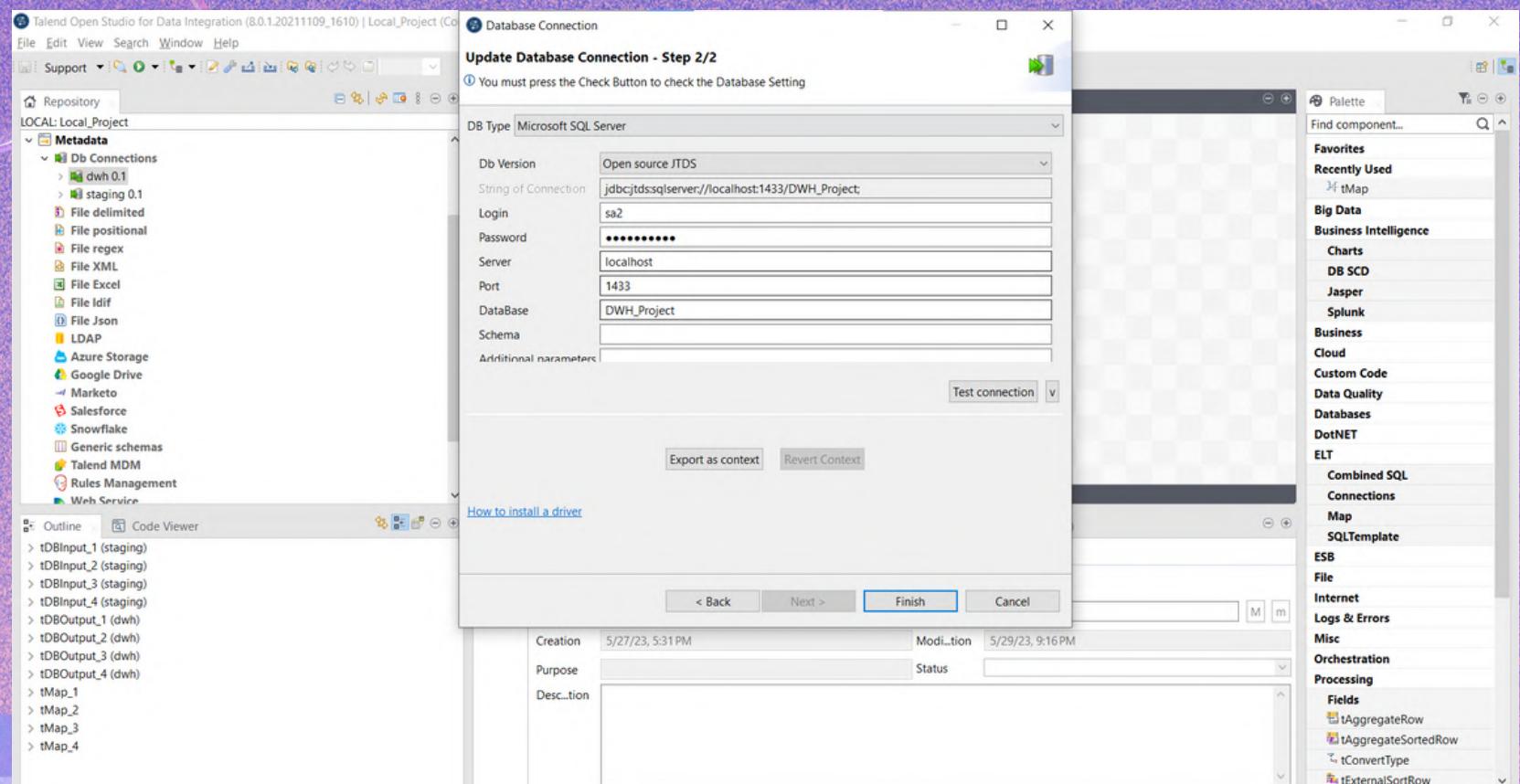
Membuat sebuah Database bernama DWH_Project, serta membuat Tabel Fact dan Dimension dari tabel yang ada di database Staging

Ketika semua Key telah dibuat jangan lupa hubungkan semua tabel untuk membuat relationships di databse diagram lalu new database diagram.



Membuat Job ETL di aplikasi talend untuk memindahkan data dari Staging ke Data Warehouse

Membuat job ETL di aplikasi talend untuk memindahkan data dari data staging ke data warehouse dengan ketentuan yang diinginkan client. Pertama-tama koneksi database yang ingin di ETL di bagian metadata -> Db connections -> lalu Create Connections, lalu setelah disetting lakukan test connections untuk mengecek apakah database terhubung atau tidak, dan jika sukses klik oke dan buat koneksi



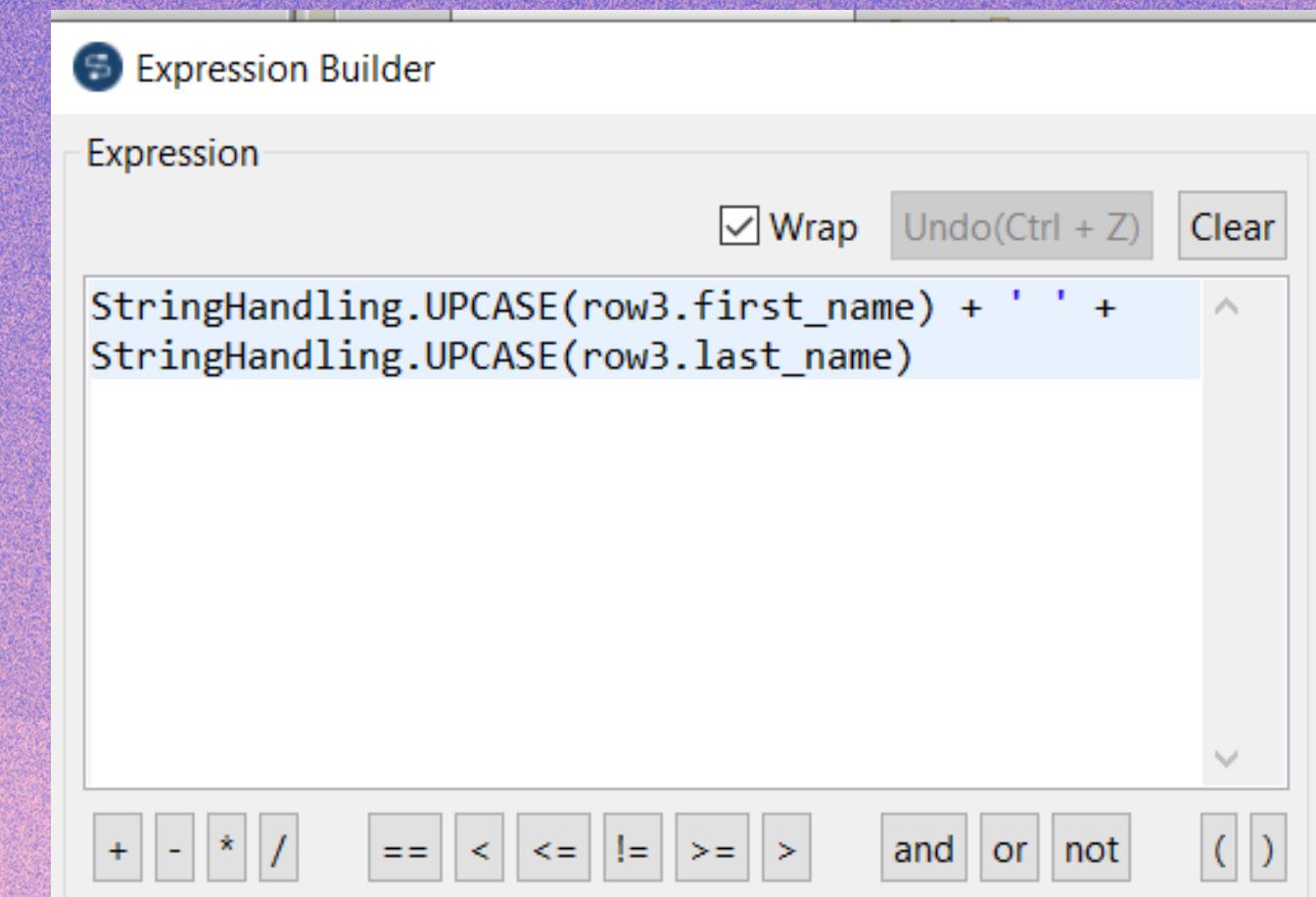
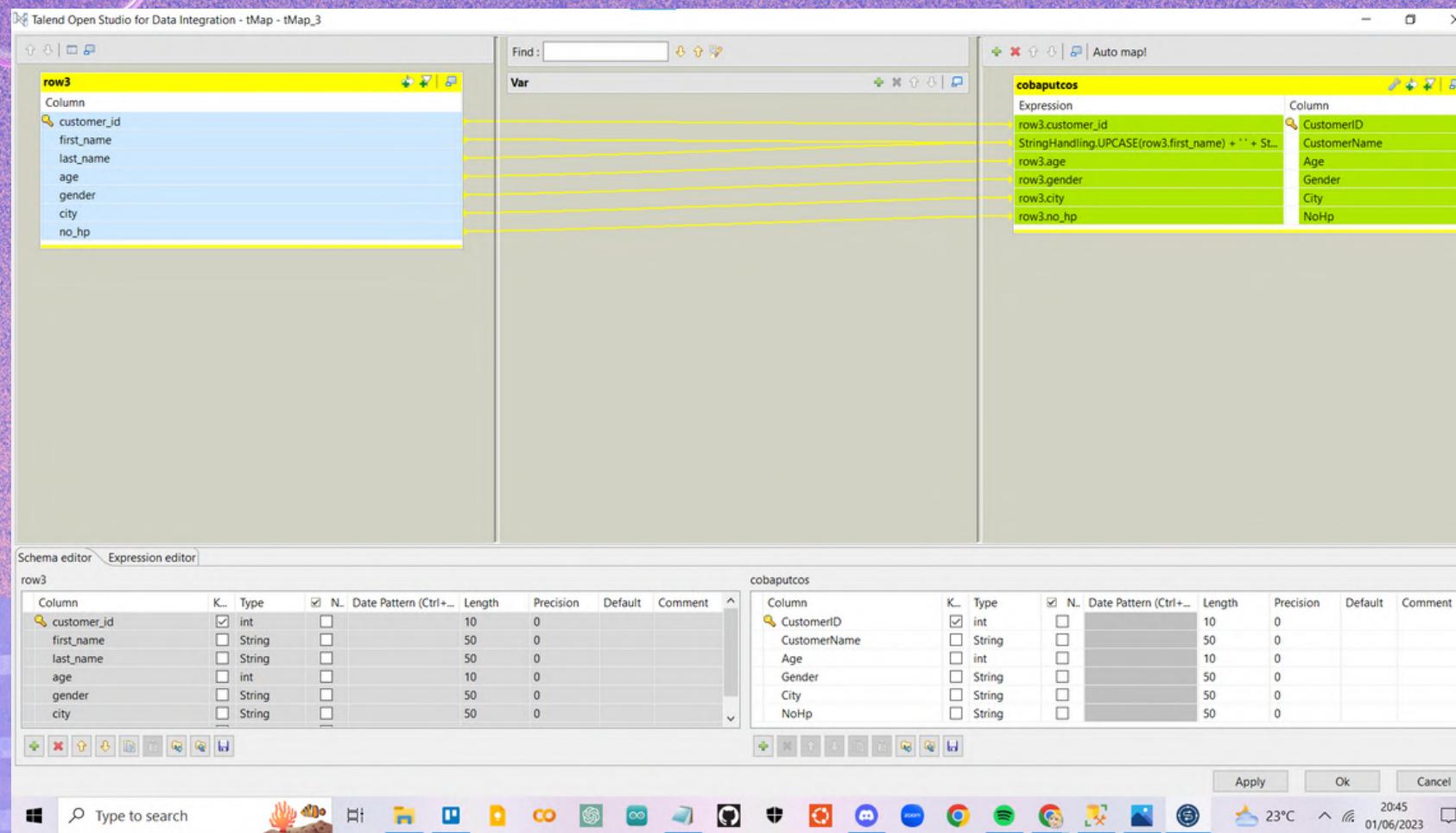
Membuat Job ETL di aplikasi talend untuk memindahkan data dari Staging ke Data Warehouse

lalu Buat job di talend menggunakan komponen tMSSqlInput dan tMSSqlOutput untuk memindahkan data dari source (staging) ke target (DWH_Project). Namun, khusus untuk tabel DimCustomer, tambah satu komponen lagi yaitu tMap untuk melakukan transformasi sesuai dengan ketentuan soal. Di dalam tMap pakai fungsi UPCASE untuk membuat huruf menjadi kapital semua, lalu untuk menggabungkan kedua kolom bisa pakai operator plus (+) . fungsi-fungsi tersebut sudah tersedia di dalam komponen tMap



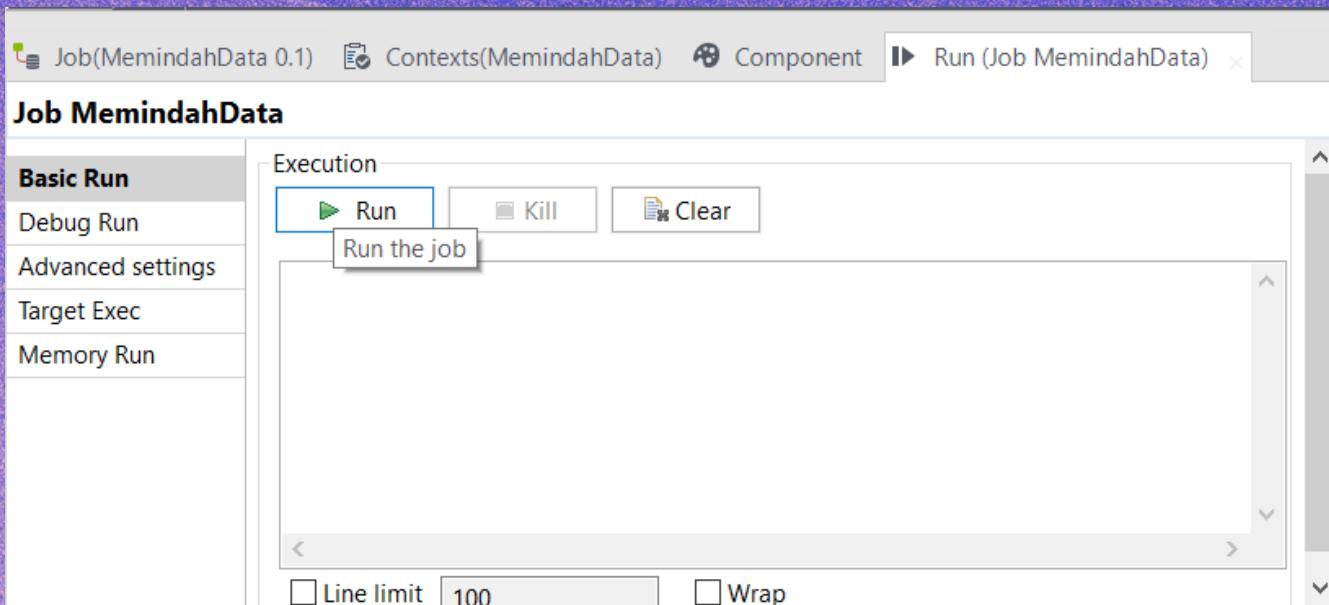
Membuat Job ETL di aplikasi talend untuk memindahkan data dari Staging ke Data Warehouse

lalu Buat job di talend menggunakan komponen tMSSqlInput dan tMSSqlOutput untuk memindahkan data dari source (staging) ke target (DWH_Project). Namun, khusus untuk tabel DimCustomer, tambah satu komponen lagi yaitu tMap untuk melakukan transformasi sesuai dengan ketentuan soal. Di dalam tMap pakai fungsi UPCASE untuk membuat huruf menjadi kapital semua, lalu untuk menggabungkan kedua kolom bisa pakai operator plus (+) . fungsi-fungsi tersebut sudah tersedia di dalam komponen tMap



Membuat Job ETL di aplikasi talend untuk memindahkan data dari Staging ke Data Warehouse

Dan untuk eksekusi job ETL silahkan klik run dibagian execution, setelah selesai silahkan cek di SSMS apakah data nya berhasil masuk dan berhasil tertransform sesuai keinginan



The screenshot displays four Microsoft SQL Server Management Studio (SSMS) windows side-by-side, each showing data from a different table:

- FactSalesOrder:** Shows sales orders with columns: OrderID, CustomerID, ProductID, Quantity, Amount, StatusID, and OrderDate. Some rows show status like 'Completed' and 'Cancelled'.
- DimStatusOrder:** Shows order statuses with columns: StatusID, StatusOrder, and StatusOrderName. Statuses include 'Awaiting Pa...', 'Menunggu ...', 'Shipped', 'Sedang Diki...', 'Completed', 'Pesanan sa...', and 'Cancelled'.
- DimCustomer:** Shows customer details with columns: CustomerID, CustomerName, Age, Gender, City, and NoHp. Examples include BUDI SANT..., AIENG SRIA..., and RIFKI MUJAH....
- DimProduct:** Shows product details with columns: ProductID, ProductName, ProductCategory, and ProductUnit. Examples include Macbook Air, T-Shirt Polo, and Bagpack Na...

Yellow arrows point from the text 'Dan untuk eksekusi job ETL silahkan klik run dibagian execution, setelah selesai silahkan cek di SSMS apakah data nya berhasil masuk dan berhasil tertransform sesuai keinginan' to the top of each SSMS window.

Membuat Store Procedure (SP) untuk menampilkan summary sales order berdasarkan status pengiriman.

Membuat Store Procedure (SP) bernama `summary_order_status` yang berisi perintah `SELECT` dan lakukan `JOIN` antara tabel fact dan dimension untuk menampilkan beberapa kolom yaitu `OrderID`, `CustomerName`, `ProductName`, `Quantity`, `StatusOrder`. Buatlah parameter bernama `StatusID` dari kolom `StatusID` (tabel `DimStatusOrder`) sebagai filter dari query tersebut. Jadi ketika kita ingin dijalankan SP tersebut sintaksnya seperti ini `EXEC summary_order_status @StatusID = ???`

```
storeprocedure.sql...H_Project (sa2 (74))  ✎ X
DROP PROCEDURE summary_order_status

CREATE PROCEDURE summary_order_status
(@StatusID int) AS
BEGIN
    SELECT
        f.OrderID,
        c.CustomerName,
        p.ProductName,
        f.Quantity,
        s.StatusOrder
    FROM FactSalesOrder f
    JOIN DimCustomer c ON f.CustomerID = c.CustomerID
    JOIN DimProduct p ON f.ProductID = p.ProductID
    JOIN DimStatusOrder s ON f.StatusID = s.StatusID

    WHERE s.StatusID = @StatusID
END

EXEC summary_order_status @StatusID = 1
```

Membuat Store Procedure (SP) untuk menampilkan summary sales order berdasarkan status pengiriman.

Terakhir Eksekusi query tersebut dan lihat apakah sudah sesuai keinginan client

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top bar indicates the file is named 'storeprocedure.sql' and is part of the 'H_Project' database, with 74 rows selected. The main pane displays the following T-SQL code:

```
(@StatusID int) AS
BEGIN
    SELECT
        f.OrderID,
        c.CustomerName,
        p.ProductName,
        f.Quantity,
        s.StatusOrder
    FROM FactSalesOrder f
    JOIN DimCustomer c ON f.CustomerID = c.CustomerID
    JOIN DimProduct p ON f.ProductID = p.ProductID
    JOIN DimStatusOrder s ON f.StatusID = s.StatusID

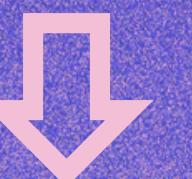
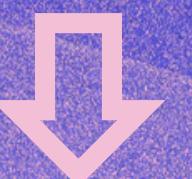
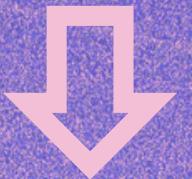
    WHERE s.StatusID = @StatusID
END

EXEC summary_order_status @StatusID = 4
```

The bottom pane shows the 'Results' tab with the following data:

	OrderID	CustomerName	ProductName	Quantity	StatusOrder
1	1302	RIFKI MUHAMMAD	HP Elitebook 840 G4	1	Completed
2	1306	RIFKI MUHAMMAD	Asus Zenbook 800	1	Completed
3	1309	AJENG SRIASIH	Bagpack Navy Club	1	Completed

Link Video and Other File Here



https://drive.google.com/drive/folders/1kGwX97bposPR7_Emtqsa5u2iMzk9JrU?usp=sharing

Conclusion

Data warehouse adalah suatu sistem yang dirancang untuk menyimpan, mengelola, dan menganalisis data yang berasal dari berbagai sumber yang berbeda. Data warehouse memungkinkan pengumpulan, pengorganisasian, dan penggabungan data yang beragam dari berbagai sumber yang terpisah, seperti basis data operasional, sistem eksternal, dan sumber data lainnya.

Tujuan utama dari data warehouse adalah untuk menyediakan basis data terpusat yang digunakan untuk analisis dan pengambilan keputusan.

Thank You

By : fahernkhan

Github : fahernkhan

linkedin : <https://www.linkedin.com/in/fafa-designer/>

