

**LAPORAN UJIAN AKHIR SEMESTER**  
***DATA WAREHOUSE***  
**SIB 2A/KELOMPOK 2**



**Disusun Oleh:**

Athallah Ayudya Paramesti	(2341760061)
Kanaya Abdielaramadhani H.	(2341760118)
Muhammad Audric Andhika	(2341760094)
Muhammad Ircham	(2341760115)
Yonanda Mayla Rusdiaty	(2341760184)

**JURUSAN TEKNOLOGI INFORMASI**  
**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**  
**POLITEKNIK NEGERI MALANG**  
**2025**

- Link Dataset : [inventory-management-dataset](#)
- Studi Kasus : Identifikasi Produk Paling Menguntungkan  
Pemetaan Distribusi Barang Berdasarkan Lokasi Gudang  
Analisis Produk dengan Perputaran Stok Tertinggi
- Tujuan : Mengidentifikasi produk yang paling menguntungkan, memetakan distribusi barang berdasarkan lokasi gudang, dan menganalisis produk dengan perputaran stok tertinggi berdasarkan data inventaris, seperti jumlah stok, tanggal pesanan, kategori produk, lokasi gudang, biaya perolehan, harga jual, dan volume penjualan historis.
- Manfaat : 1. Membantu mengurangi kerugian akibat produk tidak terjual dengan baik melalui identifikasi produk berisiko rusak, kedaluwarsa, atau usang, serta pengelolaan stok produk yang lambat terjual.  
2. Membantu dalam mengatur tata letak produk yang dirasa kurang menguntungkan dan mengoptimalkan penempatan barang di gudang untuk efisiensi distribusi.  
3. Mempermudah dalam mengatur jumlah produk yang akan distok kedepannya dengan lebih akurat, memastikan ketersediaan produk favorit, dan mengurangi biaya penyimpanan.

#### A. Perancangan Star Schema

##### 1. Tabel Fakta (Fact\_Inventory):

Tabel Fakta akan menjadi pusat dari skema bintang ini, menyimpan Foreign Keys yang menghubungkan ke tabel dimensi. Setiap baris dalam tabel fakta ini akan merepresentasikan sebuah snapshot atau peristiwa terkait status inventaris pada waktu tertentu. berikut merupakan tabel fakta yang akan kami gunakan pada Ujian Akhir semester ini :

##### A. Produk menguntungkan:

- fakta\_persediaan

##### B. Distribusi barang:

- fakta\_persediaan

##### C. Perputaran stok:

- fakta\_persediaan

## 2. Tabel Dimensi :

Tabel Dimensi akan menjadi cabang-cabang pelengkap yang mengelilingi tabel fakta dalam skema bintang ini, menyediakan konteks deskriptif dan Primary Keys yang akan dihubungkan oleh foreign keys dari tabel fakta. Setiap baris dalam tabel dimensi ini akan merepresentasikan entitas unik seperti produk, waktu, atau lokasi gudang, lengkap dengan atribut-atribut penjelas yang memberikan pemahaman mendalam tentang setiap peristiwa inventaris yang tercatat. berikut merupakan tabel dimensi yang akan kami gunakan pada Ujian Akhir semester ini

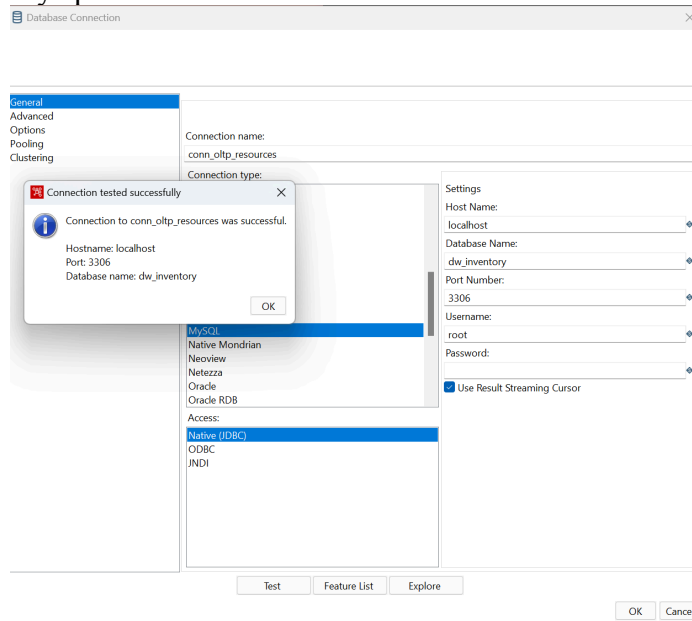
- a) Produk menguntungkan:
  - dim\_produk
- b) Distribusi barang:
  - dim\_gudang, dim\_produk
- c) Perputaran stok:
  - dim\_produk

## B.Langkah-Langkah Implementasi Transformasi Data Menggunakan Pentaho

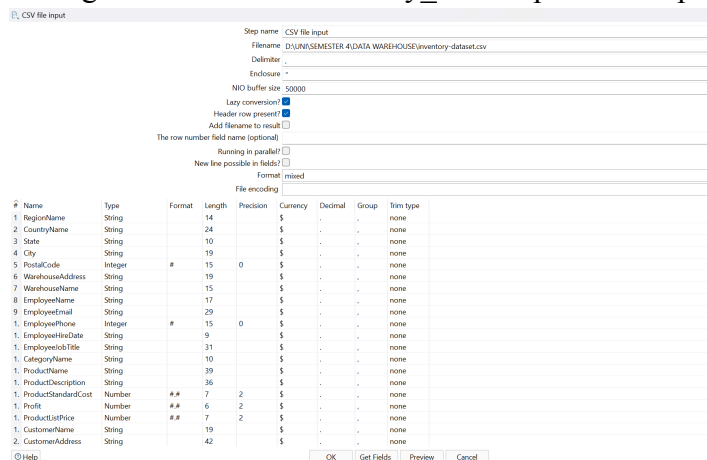
Dalam pengolahan data warehouse, kami menggunakan Pentaho Data Integration untuk mentransformasi dan memuat data ke dalam skema bintang (star schema) yang terdiri dari tabel fakta dan tabel dimensi. Berikut adalah langkah-langkah implementasinya secara bertahap berdasarkan setiap bagian visualisasi proses transformasi.

### A. Dim\_produk:

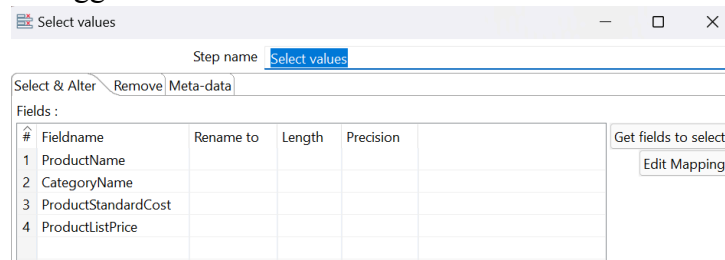
1. lakukan koneksi database sesuai dengan nama database yang telah dibuat di mysql



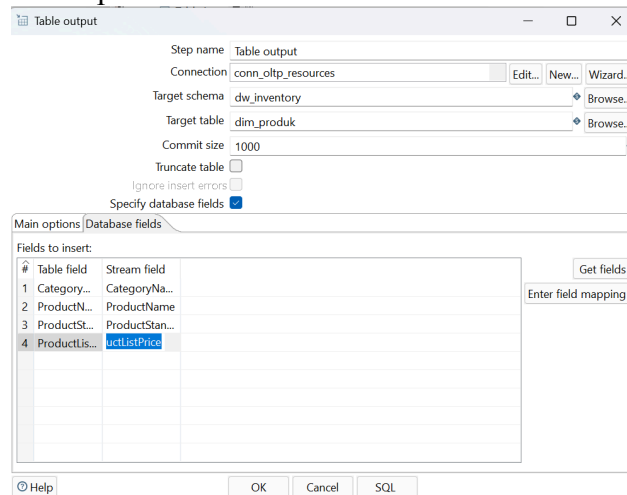
2. mengambil data dari inventory\_dataset pada csv input



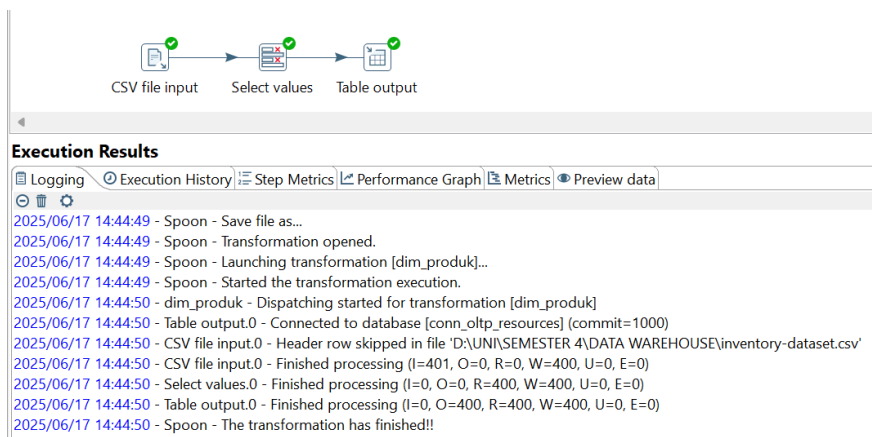
3. menyeleksi kolom-kolom yang sekiranya tidak dibutuhkan dan melakukan penstandaran nama kolom agar sesuai dengan struktur database dengan menggunakan “select values”



4. mengarahkan output ke table dim\_produk didalam database, membuat id\_produk sebagai Primary key dan untuk sisanya digunakan sebagai atribut deskriptif



5. melihat hasil dari dim\_produk, gambar berikut menunjukkan bahwa data dim\_produk berhasil masuk ke dalam database

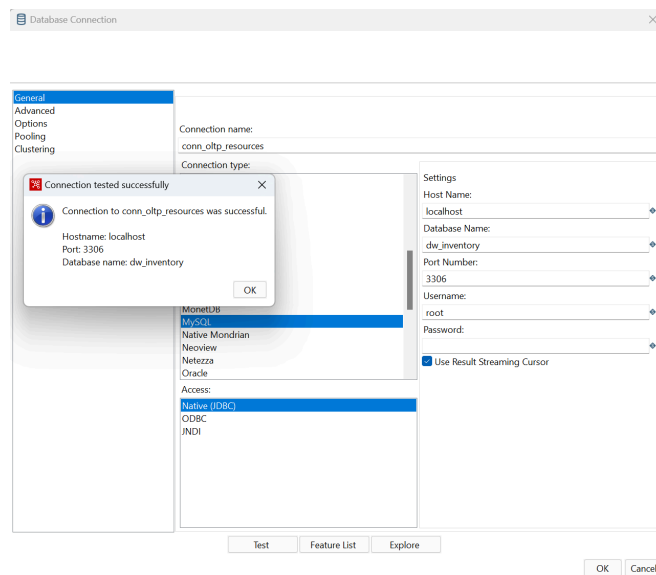


## Hasil Eksekusi dim\_produk:

Execution Results				
Logging Execution History Step Metrics Performance Graph Metrics Preview data				
First rows Last rows Off				
#	ProductName	CategoryName	ProductStandardCost	ProductListPrice
1	Intel Xeon E5-2699 V3 (OEM/Tray)	CPU	2867.51	3410.46
2	Intel Xeon E5-2697 V3	CPU	2326.27	2774.98
3	Intel Xeon E5-2698 V3 (OEM/Tray)	CPU	2035.18	2660.72
4	Intel Xeon E5-2697 V4	CPU	2144.4	2554.99
5	Intel Xeon E5-2685 V3 (OEM/Tray)	CPU	2012.11	2501.69
6	Intel Xeon E5-2695 V3 (OEM/Tray)	CPU	1925.13	2431.95
7	Intel Xeon E5-2697 V2	CPU	2101.59	2377.09
8	Intel Xeon E5-2695 V4	CPU	1780.35	2269.99
9	Intel Xeon E5-2695 V2	CPU	1793.53	2259.99
1.	Intel Xeon E5-2643 V2 (OEM/Tray)	CPU	1940.18	2200.0
1.	Intel Xeon E5-2690 (OEM/Tray)	CPU	1888.33	2116.72
1.	Intel Xeon E5-2687W V3	CPU	1781.47	2064.99
1.	Intel Xeon E5-2687W V4	CPU	1723.83	2042.69
1.	Intel Xeon E5-2667 V3 (OEM/Tray)	CPU	1504.08	2009.46
1.	Intel Xeon E5-2690 V4	CPU	1499.26	1994.49
1.	Intel Xeon E5-2690 V3	CPU	1540.35	1908.73
1.	Intel Xeon E5-2470V2	CPU	1671.95	1904.7
1.	Intel Xeon E5-2683 V4	CPU	1706.95	1899.99

## B. dim\_gudang:

1. lakukan koneksi database sesuai dengan nama database yang telah dibuat di mysql



## 2. mengambil data dari inventory\_dataset pada csv input

CSV file input

Step name: CSV file input

Filename: D:\UNI\SEMESTER 4\DATA WAREHOUSE\inventory-dataset.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	RegionName	String		14		\$	.	.	none
2	CountryName	String		24		\$	.	.	none
3	State	String		10		\$	.	.	none
4	City	String		19		\$	.	.	none
5	PostalCode	Integer	#	15	0	\$	.	.	none
6	WarehouseAddress	String		19		\$	.	.	none
7	WarehouseName	String		15		\$	.	.	none
8	EmployeeName	String		17		\$	.	.	none
9	EmployeeEmail	String		29		\$	.	.	none
1.	EmployeePhone	Integer	#	15	0	\$	.	.	none
1.	EmployeeHireDate	String		9		\$	.	.	none
1.	EmployeeJobTitle	String		31		\$	.	.	none
1.	CategoryName	String		10		\$	.	.	none
1.	ProductName	String		39		\$	.	.	none
1.	ProductDescription	String		36		\$	.	.	none
1.	ProductStandardCost	Number	##	7	2	\$	.	.	none
1.	Profit	Number	##	6	2	\$	.	.	none
1.	ProductListPrice	Number	##	7	2	\$	.	.	none
1.	CustomerName	String		19		\$	.	.	none

Help OK Get Fields Preview Cancel

## 3. menyeleksi kolom-kolom yang sekiranya tidak dibutuhkan dan melakukan penstandaran nama kolom agar sesuai dengan struktur database dengan menggunakan "select values"

Select values

Step name: Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	WarehouseName			
2	WarehouseAddress			
3	City			
4	State			

Get fields to select

Edit Mapping

- 
- Table output
- Step name: **table\_output**
- Connection: **conn\_olap\_resource** [Edit...] [New...] [Wizards...]
- Target schema: **dw\_inventory** [Browse...]
- Target table: **dim\_gudang** [Browse...]
- Commit size: **1000**
- Truncate table: ☐
- Ignore insert errors: ☐
- Specify database fields: ☒
- Main options | Database fields
- Partition data over tables: ☐
- Partitioning field: [Dropdown menu]
- Partition data per month: ☒
- Partition data per day: ☐
- Use batch update for inserts: ☒
- Is the name of the table defined in a field?: ☐
- Field that contains name of table: [Dropdown menu]
- Store the tablename field: ☒
- Return auto-generated key: ☐
- Name of auto-generated key field: [Text field]
- [Help] [OK] [Cancel] [SQL]

Diagram illustrating the data flow: CSV file input → Select values → Table output.

Diagram illustrating the data flow process:

```

graph LR
    A[CSV file input] --> B[Select values]
    B --> C[Table output]
  
```

The process starts with a **CSV file input** (represented by a document icon with a green checkmark), which flows into **Select values** (represented by a table icon with a green checkmark and a red 'X' indicating a selection operation). The output of the selection process is a **Table output** (represented by a table icon with a green checkmark).

## Execution Results

Logging ○ Execution History **Step Metrics** Performance Graph Metrics Preview data

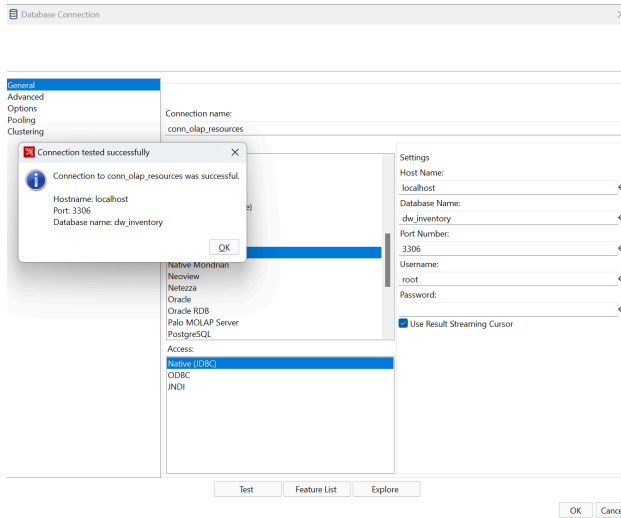
☒ First rows ☐ Last rows ☐ Off

#	WarehouseName	WarehouseAddress	City	State
1	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
2	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
3	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
4	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
5	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
6	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
7	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
8	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
9	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
1.	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas
×	Southlake Texas	2014 Jabberwocky Rd	Southlake	Texas

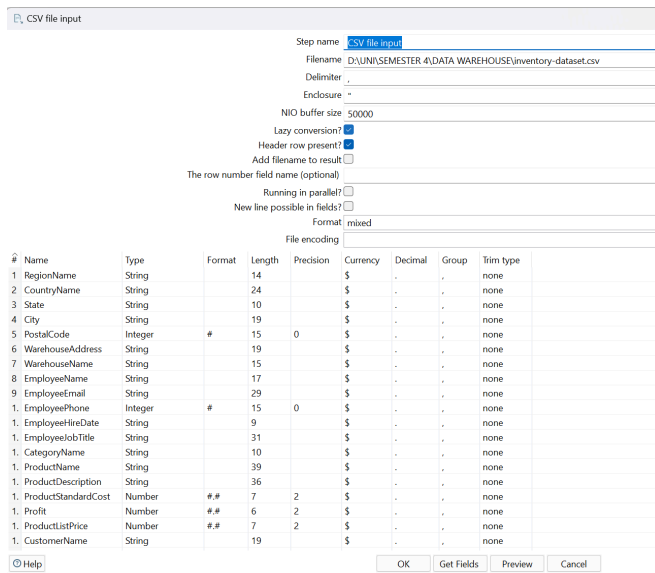


### C. tabel\_fakta:

1. lakukan koneksi database sesuai dengan nama database yang telah dibuat di mysql



2. Mengambil data transaksi Utama seperti id\_produk, id\_gudang, stok, tanggal\_transaksi, volume\_penjualan.



3. Product Lookup:  
Mencari id\_produk ke dim\_produk untuk mengambil detail seperti kategori/harga.

The screenshot shows two configuration windows. The top window, titled 'Table input', has 'Step name' set to 'produk lookup' and 'Connection' set to 'conn\_olap\_resources'. The SQL query is 'Select productID, CategoryName, ProductName, ProductStandardCost, ProductListPrice From dim\_produk'. The bottom window, titled 'Stream lookup', has 'Step name' set to 'lookup produk' and 'Lookup step' set to 'produk lookup'. It lists four key fields: ProductName, CategoryName, ProductStandardCost, and ProductListPrice. The 'Specify the fields to retrieve' section shows 'productID' as the field to retrieve, with a 'New name' of 'productID' and a 'Type' of 'None'.

#	Field	LookupField
1	ProductName	ProductName
2	CategoryName	CategoryName
3	ProductStandardCost	ProductStandardCost
4	ProductListPrice	ProductListPrice

#	Field	New name	Default	Type
1	productID			None

4. Gudang lookup dan lookup Gudang:  
Lookup dari id\_gudang ke dim\_gudang untuk mengambil nama dan lokasi gudang.

The screenshot shows two configuration windows. The top window, titled 'Table input', has 'Step name' set to 'gudang lookup' and 'Connection' set to 'conn\_olap\_resources'. The SQL query is 'SELECT WarehouseID, WarehouseName, WarehouseAddress, City, State FROM dim\_gudang'. The bottom window, titled 'Stream lookup', has 'Step name' set to 'lookup gudang' and 'Lookup step' set to 'gudang lookup'. It lists four key fields: WarehouseName, WarehouseAddress, City, and State. The 'Specify the fields to retrieve' section shows 'WarehouseID' as the field to retrieve, with a 'New name' of 'WarehouseID' and a 'Type' of 'None'.

#	Field	LookupField
1	WarehouseName	WarehouseName
2	WarehouseAddress	WarehouseAddress
3	City	City
4	State	State

#	Field	New name	Default	Type
1	WarehouseID			None

5. select value:  
Memilih kolom yang akan dimasukkan ke tabel fakta seperti:  
id\_produk, id\_gudang, stok, volume\_penjualan, harga\_beli, harga\_jual.

#	Fieldname	Rename to	Length	Precision
1	productID			
2	WarehouseID			
3	OrderItemQuantity			
4	PerUnitPrice			
5	Profit			

6. Table Output:  
berikan output ke tabel fakta\_persediaan di database

Step name: Table output

Connection: conn\_olap\_resources

Target schema: dw\_inventory

Target table: fakta\_persediaan

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options: Database fields

Partition data over tables: ☐

Partitioning field:

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table:

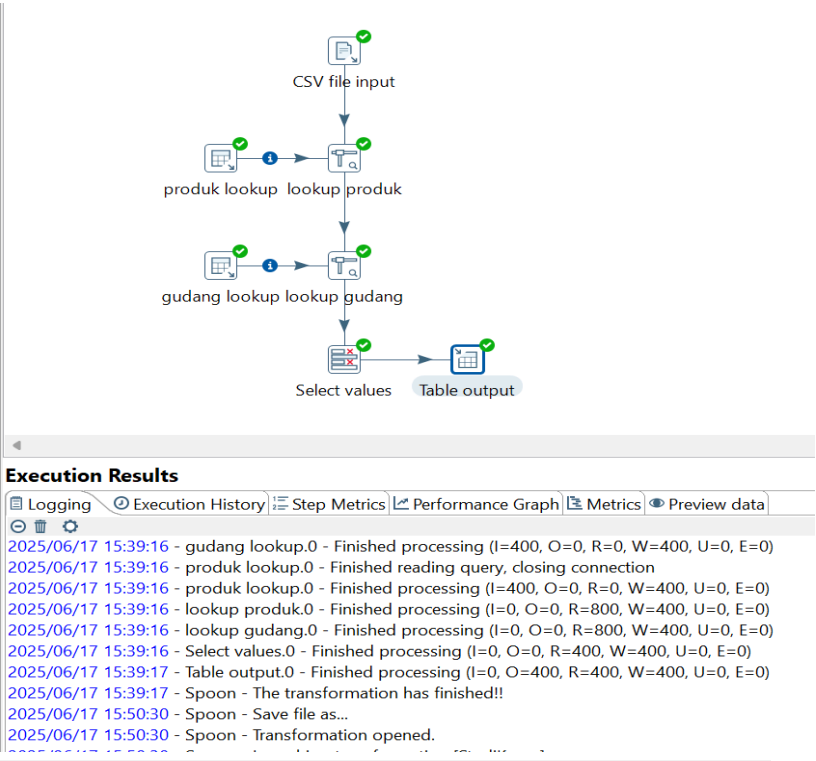
Store the tablename field: ☐

Return auto-generated key: ☐

Name of auto-generated key field:

Buttons: Help, OK, Cancel, SQL

7. Hasil:



**Execution Results**

Logging Execution History Step Metrics Performance Graph Metrics Preview data

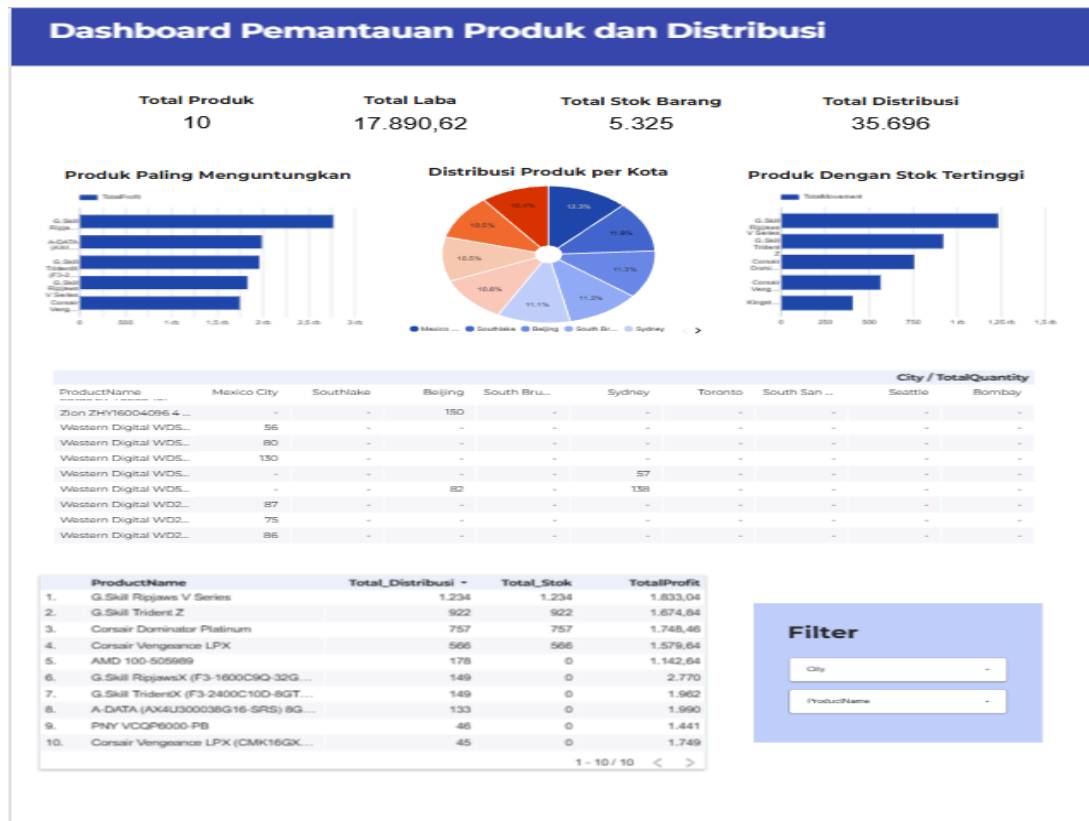
☒ First rows ☐ Last rows ☐ Off

#	productID	WarehouseID	OrderItemQuantity	PerUnitPrice	Profit
1	1	44	132	469.99	542.95
2	2	44	124	519.99	448.71
3	3	44	92	800.74	625.54
4	4	44	128	849.99	410.59
5	5	44	106	109.99	489.58
6	6	44	148	549.59	506.82
7	321	44	127	659.99	275.5
8	322	44	138	54.99	489.64

### C.Penerapan Looker Studio:

Pada tahap ini, data yang telah diolah dan dimuat ke dalam skema data warehouse digunakan untuk membuat visualisasi yang lebih user friendly dengan menggunakan Looker Studio . Visualisasi ini bertujuan untuk menampilkan hasil analisis seperti produk paling menguntungkan, distribusi barang berdasarkan lokasi gudang, dan perputaran stok tertinggi secara lebih informatif dan mudah dipahami oleh pengguna non-teknis.

Dengan menghubungkan database ke Looker Studio, pengguna dapat membangun dashboard yang menampilkan grafik, tabel secara real-time. Dashboard ini mendukung pengambilan keputusan bisnis yang lebih cepat dan rapi. berikut merupakan dashboard yang telah dibuat di lookerstudio.



Untuk detail yang lebih komprehensif dapat disimak pada link berikut:

<https://lookerstudio.google.com/reporting/90331f97-1a34-4343-b5c1-43c02b87bc44>