PROJECT KELOMPOK
RETAIL/E-COMMERCE SALES ANALYTICS
BUSINESS INTELLIGENCE

Oleh:

Agung Fradiansyah	03	2341720045
Ahmad Ramadhan Baiakbar	04	2341720085
Febriansyah adi nugroho	12	2341720023
Pramudya Surya Anggara Putra	24	2341720141

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

1.1 SUMBER DATA

Data yang digunakan dalam proyek ini diperoleh dari platform **Kaggle** dengan judul dataset **"Superstore Sales Dataset"**. Dataset tersebut dapat diakses secara legal dan gratis melalui tautan berikut:

Link Dataset:

https://www.kaggle.com/datasets/rohitsahoo/sales-forecasting/data

Alasan Pemilihan Dataset

Dataset ini dipilih karena memenuhi beberapa kriteria yang diperlukan untuk pengembangan proyek ETL dan data warehouse, yaitu:

1. Data lengkap dan terstruktur, sehingga memudahkan proses ekstraksi, transformasi, dan pemuatan.
2. Relevan dengan kebutuhan analisis, khususnya dalam konteks penjualan, pelanggan, produk, dan transaksi.
3. Akses legal dan gratis, sehingga aman digunakan untuk tujuan akademik maupun praktikum.

1.2 DESKRIPSI STUDI KASUS

Dataset yang digunakan dalam proyek ini berasal dari platform **Kaggle** dengan judul **"Superstore Sales Dataset"**, yang berisi data transaksi penjualan dari sebuah perusahaan retail fiktif bernama Global Superstore. Dataset ini mencakup informasi lengkap mengenai pesanan, pelanggan, produk, lokasi, serta nilai penjualan, sehingga sangat sesuai untuk kebutuhan analisis bisnis dan pembangunan data warehouse.

Dalam dataset ini tersedia berbagai atribut penting seperti**:**

1. **Order ID**

   Merupakan kode unik untuk setiap transaksi yang terjadi. Atribut ini membantu kita membedakan satu pesanan dengan pesanan lainnya.

2. **Order Date**

   Tanggal ketika pelanggan melakukan pemesanan. Data ini nantinya berguna untuk melihat pola atau tren penjualan berdasarkan waktu.

3. **Ship Date**

   Tanggal barang dikirim ke pelanggan. Atribut ini dapat digunakan untuk menghitung lama waktu pengiriman serta mengevaluasi kinerja logistik.

4. **Ship Mode**

   Jenis layanan pengiriman yang digunakan, seperti Standard Class atau First Class. Informasi ini bisa membantu menganalisis pengaruh metode pengiriman terhadap kecepatan maupun kepuasan pelanggan.

5. **Customer ID**

   Identitas unik untuk masing-masing pelanggan. Digunakan untuk melacak riwayat pembelian dan melakukan segmentasi pelanggan.

6. **Customer Name**

   Nama pelanggan yang melakukan pembelian. Umumnya digunakan untuk identifikasi, tetapi analisis lebih sering berfokus pada Customer ID karena lebih konsisten.

7. **Segment**

   Kategori pelanggan berdasarkan jenis pengguna, misalnya Consumer, Corporate, atau Home Office. Atribut ini untuk mengetahui kontribusi penjualan dari tiap segmen

8. **Country**

   Negara tempat pelanggan berada. Dalam dataset ini biasanya bernilai sama, namun tetap dicantumkan sebagai informasi geografis

9. **City**

   Kota asal pelanggan. Data ini membantu analisis penjualan berdasarkan wilayah kota.

10. **State**

    Provinsi atau negara bagian tempat pelanggan tinggal. Atribut ini umum dipakai saat perusahaan ingin melihat performa penjualan per wilayah.

11. **Postal Code**

    Kode pos alamat pengiriman. Digunakan untuk melakukan pemetaan lokasi dengan lebih detail

12. **Region**

    Pembagian wilayah penjualan seperti East, West, Central, dan South. Atribut ini sering digunakan untuk analisis regional atau pembuatan dashboard

13. **Product ID**

Identitas unik untuk setiap produk. Atribut ini sangat penting saat menghubungkan data transaksi ke data detail produk

14. **Category**

    Kategori utama dari produk, seperti Furniture, Technology, atau Office Supplies. Biasanya menjadi dasar analisis performa tiap kategori.

15. **Sub-Category**

    Sub-kategori yang lebih spesifik dari kategori utama, misalnya Chairs, Phones, Binders, dan lain-lain. Memungkinkan analisis yang lebih mendalam terhadap jenis produk tertentu.

16. **Product Name**

    Nama lengkap produk yang dibeli oleh pelanggan. Berguna untuk mengenali produk secara spesifik dalam laporan dan analisis.

17. **Sales**

    Jumlah pendapatan yang dihasilkan dari transaksi tersebut. Ini merupakan atribut utama yang digunakan dalam perhitungan total penjualan, KPI, serta analisis performa bisnis

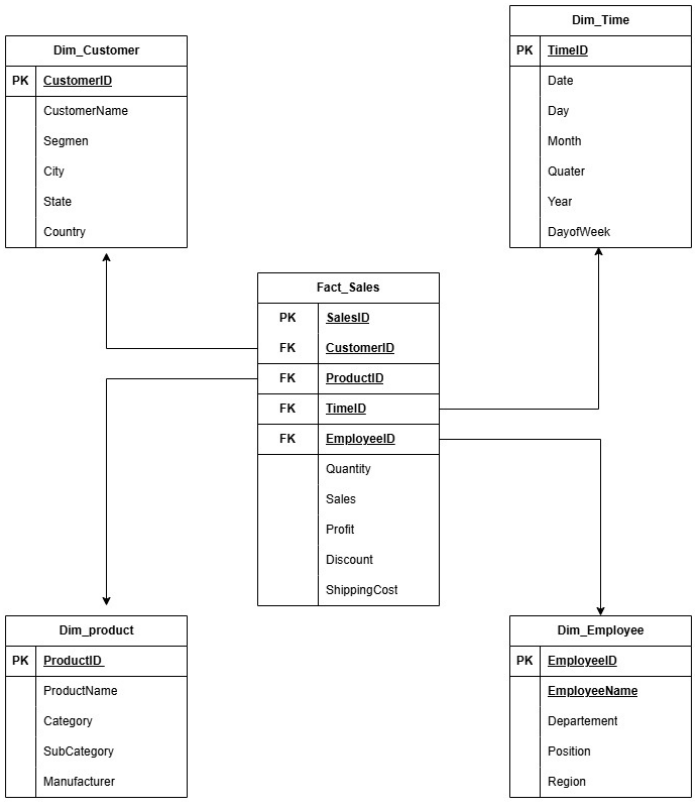
Data tersebut menggambarkan aktivitas penjualan harian selama beberapa tahun, lengkap dengan detail pelanggan serta kategori produk yang dibeli.

Kasus yang diangkat dalam proyek ini adalah bagaimana Global Superstore ingin menganalisis kinerja penjualan untuk mendukung pengambilan keputusan strategis. Analisis dilakukan untuk mengetahui tren penjualan berdasarkan waktu, produk dengan performa terbaik, segmentasi pelanggan, wilayah dengan kontribusi tertinggi, serta efektivitas metode pengiriman. Selain itu, dataset ini sangat relevan untuk digunakan dalam proses **ETL (Extract, Transform, Load)** serta perancangan **skema bintang (star schema)** dalam pembangunan data warehouse.

1.3 STAR SCHEMA (SKEMA BINTANG)

Setelah menganalisis dataset yang kami gunakan, kami menggunakan 4 tabel utama sebagai tabel dimensi dan 1 tabel sebagai tabel fakta.

Studi kasus Supert Store sales dataset:



a. Tabel fakta :

1. Fact_Sales (atribut: salesID, CustomerID, ProductID, TimeID, EmployeeID, Quantity, Sales, Profit, Discount, ShippingCost)

b. Tabel dimensi :

1. Dim_Customer (CustomerID,CustomerName, Segmen, City, State, Country)
2. Dim_Time (TimeID, Date, Day, Month, Quarter, Year, DayofWeek)
3. Dim_product (ProductID, ProductName, Category, SubCategory, Manufacturer)
4. Dim_Employee (EmployeeID, EmployeeName, Departement, Position, Region)

1.4 ETL

Setelah proses perancangan skema bintang (star schema) selesai, tahap berikutnya adalah mengimplementasikan proses **ETL (Extract, Transform, Load)**. ETL merupakan inti dari pembangunan data warehouse:

1. **Extract** yaitu untuk mengambil data dari sumber awal (dataset CSV)
2. **Transform** yaitu untuk membersihkan, mengelompokkan, menambahkan atribut, dan memastikan data sesuai kebutuhan analisis.
3. **Load** yaitu untuk memuat data ke dalam tabel dimensi dan tabel fakta di data warehouse

Pada proyek ini, proses ETL dilakukan menggunakan Pentaho Data Integration (PDI/Spoon). Proses ETL dibagi menjadi lima transformation yang masing-masing digunakan untuk mengisi tabel dimensi dan tabel fakta.

1. Dim_Customer, yaitu transformation untuk melakukan extract dan load data pelanggan.
2. Dim_Product, yaitu transformation untuk melakukan extract dan load data produk.
3. Dim_Employee, yaitu transformation untuk menghasilkan data pegawai dan memuatnya ke tabel dimensi.
4. Dim_Time, yaitu transformation untuk menghasilkan data kalender atau dimensi waktu.
5. Fact_Sales, yaitu transformation yang memuat data transaksi ke dalam tabel fakta penjualan.

1.4.1 Setup Awal: Membuat Database Connection di Pentaho

Sebelum menjalankan proses ETL, langkah pertama yang dilakukan adalah menyiapkan koneksi antara Pentaho dan database data warehouse. Koneksi ini diperlukan agar setiap transformation dapat membaca dan menulis data ke dalam tabel yang sudah dibuat pada MySQL.

Berikut langkah-langkah pembuatan koneksi database:

1. Membuka aplikasi Pentaho Spoon dengan menjalankan file Spoon.bat pada sistem operasi Windows atau spoon.sh pada Linux dan MacOS.
2. Membuat sebuah transformation baru melalui menu File kemudian memilih New dan Transformation.

3. Membuat koneksi database melalui panel View yang terdapat di sisi kiri, kemudian memilih Database Connections dan memilih opsi New.
4. Mengisi informasi koneksi pada dialog Database Connection yang muncul

   Connection Name: bi_warehouse_mysql

   Connection Type: MySQL

   Access: Native (JDBC)

   Settings tab:

   Host Name: localhost

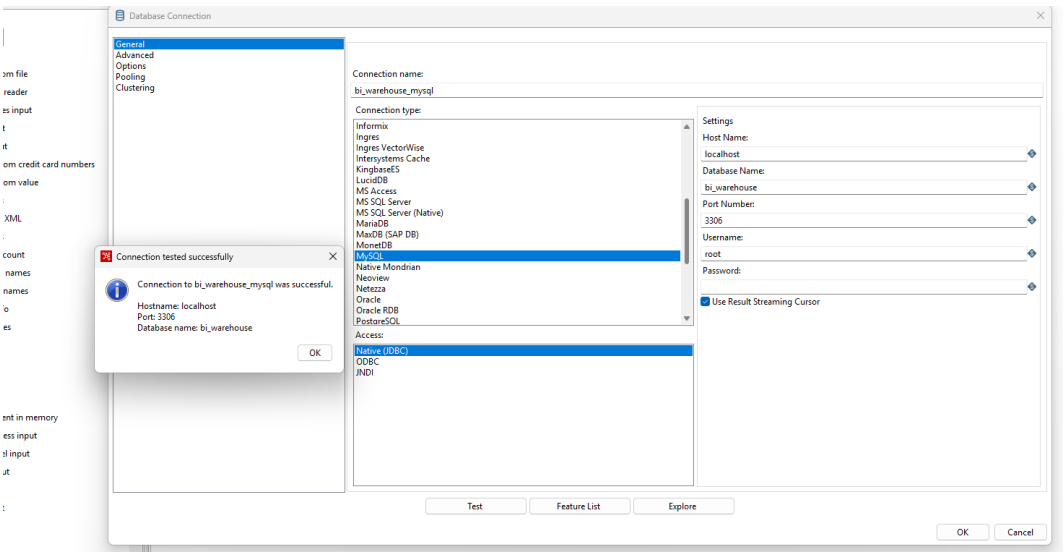
   Database Name: bi_warehouse

   Port Number: 3306

   User Name: root

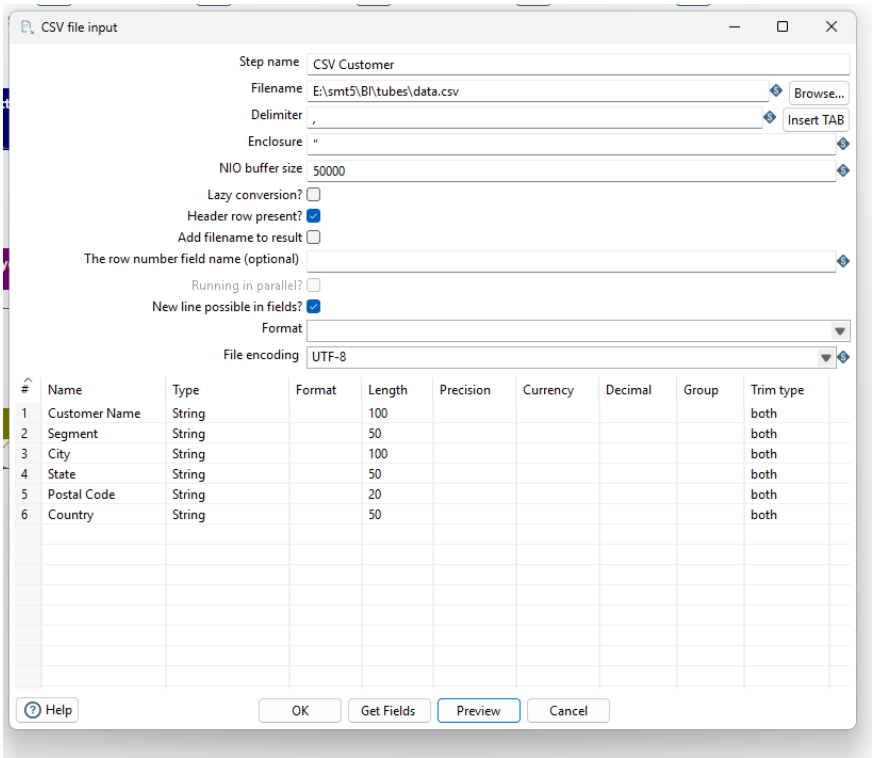
   Password:

   Advanced tab: Supports Synchronization After Statement: TRUE
5. Kemudian menguji koneksi dengan menekan tombol Test hingga muncul pesan “Connection to database 'bi_warehouse' is OK”. Jika koneksi berhasil, proses dilanjutkan dengan menekan tombol OK untuk menyimpan konfigurasi



1.4.2 Transormation 1: load Dim_Custumer

1. **Komponen CSV Input**



Cara:

1 Dari panel **Input** (kiri), drag **"CSV file input"** ke canvas

2 Double-click step untuk configure

Konfigurasi:

Pada Tab **File:**

File or directory: Browse ke data.csv

Regular Expression: (kosongkan)

Tab **Content:**

Separator: ,

Enclosure: "

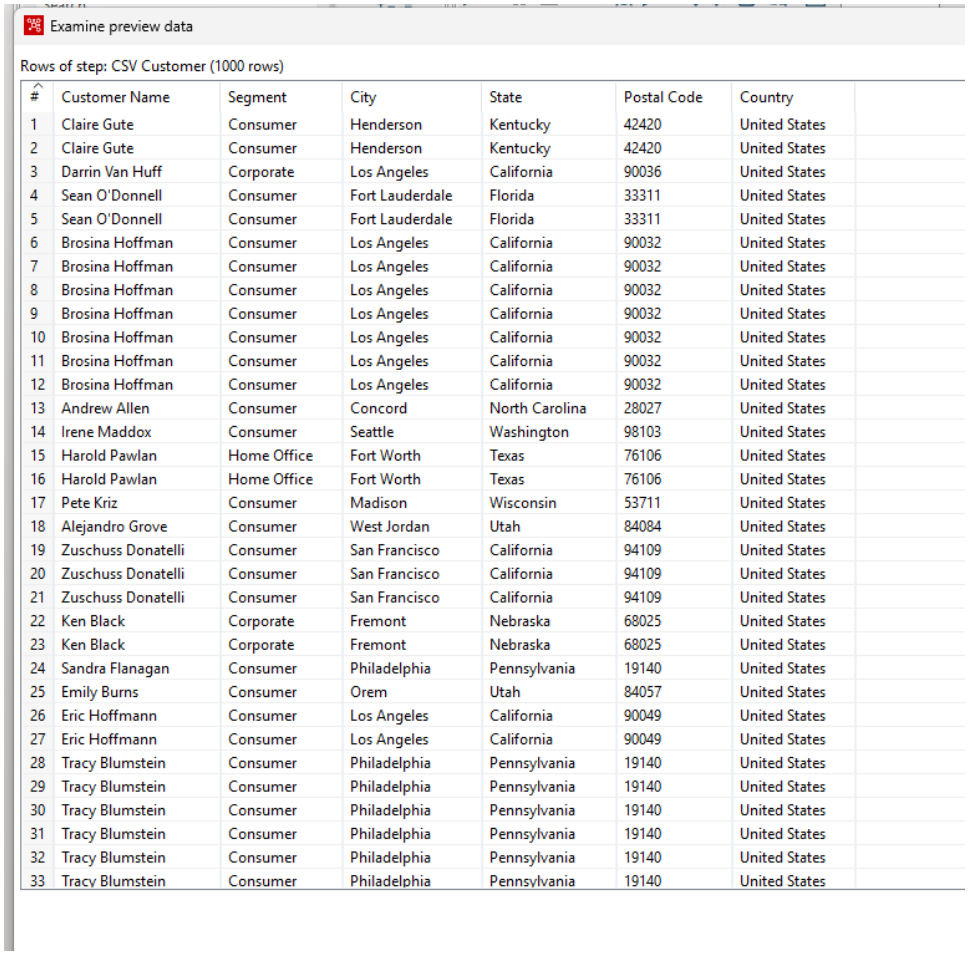
Header: Yes (centang)

Nr header lines: 1

Format: Unix

Encoding: UTF-8

Pada tab **Fields:** Klik **"Get Fields"** button untuk auto-detect semua kolom dataset. Atau klik preview maka akan tampil seperti berikut, lalu klik O



Examine preview data

Rows of step: CSV Customer (1000 rows)

#	Customer Name	Segment	City	State	Postal Code	Country
1	Claire Gute	Consumer	Henderson	Kentucky	42420	United States
2	Claire Gute	Consumer	Henderson	Kentucky	42420	United States
3	Darrin Van Huff	Corporate	Los Angeles	California	90036	United States
4	Sean O'Donnell	Consumer	Fort Lauderdale	Florida	33311	United States
5	Sean O'Donnell	Consumer	Fort Lauderdale	Florida	33311	United States
6	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
7	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
8	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
9	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
10	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
11	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
12	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
13	Andrew Allen	Consumer	Concord	North Carolina	28027	United States
14	Irene Maddox	Consumer	Seattle	Washington	98103	United States
15	Harold Pawlan	Home Office	Fort Worth	Texas	76106	United States
16	Harold Pawlan	Home Office	Fort Worth	Texas	76106	United States
17	Pete Kriz	Consumer	Madison	Wisconsin	53711	United States
18	Alejandro Grove	Consumer	West Jordan	Utah	84084	United States
19	Zuschuss Donatelli	Consumer	San Francisco	California	94109	United States
20	Zuschuss Donatelli	Consumer	San Francisco	California	94109	United States
21	Zuschuss Donatelli	Consumer	San Francisco	California	94109	United States
22	Ken Black	Corporate	Fremont	Nebraska	68025	United States
23	Ken Black	Corporate	Fremont	Nebraska	68025	United States
24	Sandra Flanagan	Consumer	Philadelphia	Pennsylvania	19140	United States
25	Emily Burns	Consumer	Orem	Utah	84057	United States
26	Eric Hoffmann	Consumer	Los Angeles	California	90049	United States
27	Eric Hoffmann	Consumer	Los Angeles	California	90049	United States
28	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States
29	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States
30	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States
31	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States
32	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States
33	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States

2. Select Values - Pilih Customer Fields

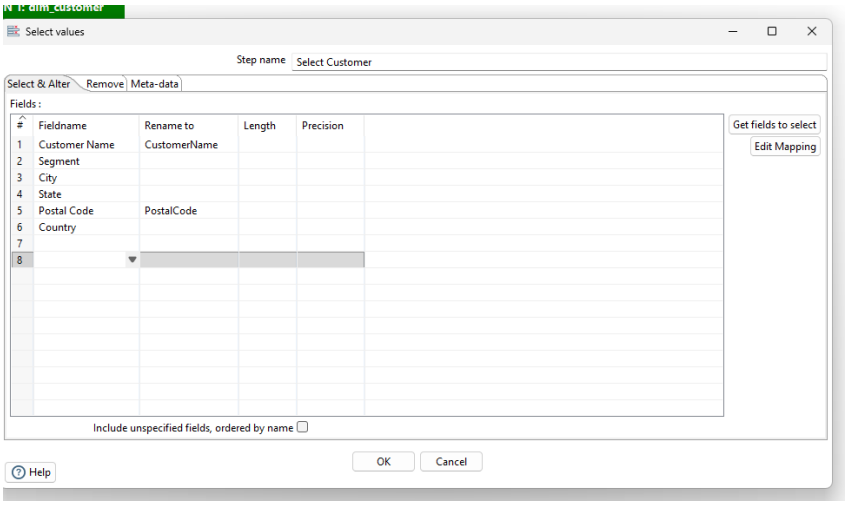
Komponen **Select Values** digunakan untuk memilih kolom tertentu dari dataset agar hanya data yang relevan saja yang diteruskan ke proses berikutnya.

Cara:

1 Drag **"Select values"** dari panel **Transform**

2 Connect: CSV File Input ke Select Values

3 Double-click Select Values untuk configure, seperti berikut, Lalu klik OK

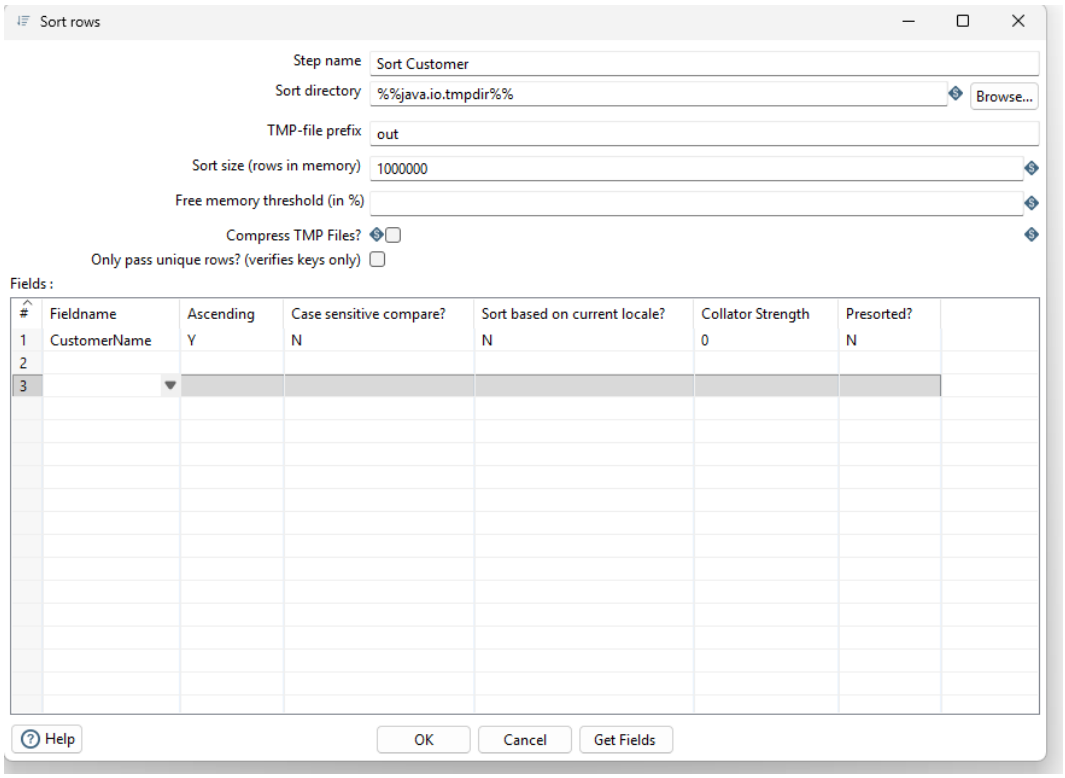


3. Sort Rows - Sort by Customer ID

Cara:

- Drag **"Sort rows"** dari **Transform**
- Connect: Select Values ke Sort rows

Konfigurasi:



a. Fieldname: customer_id

b. Sort direction: Ascending

c. Case sensitive: No

d. Pre-sort rows: NO

4. **Unique Rows - Keep Unique Customers**

Komponen ini diigunakan untuk menghapus data duplikat agar setiap Customer ID hanya muncul satu kali di tabel Dim_Customer

Cara:

1. Drag **"Unique rows"** dari **Transform**
2. Connect: Sort rows ke Unique rows

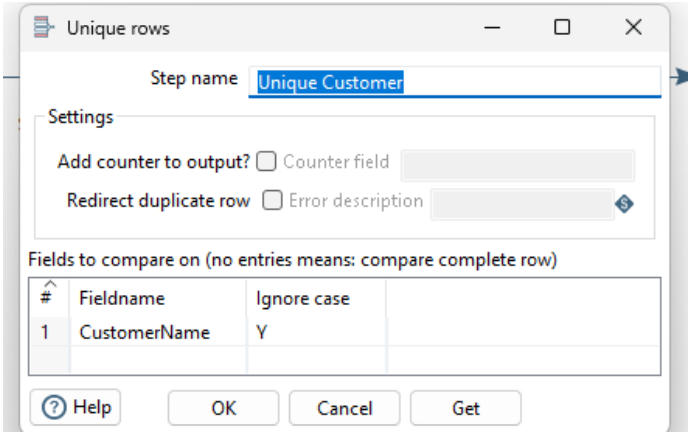
Konfigurasi:

Redirect duplicate row: NO (uncheck)

Error on duplicate: NO (uncheck)

Count rows: (kosongkan)

Count field: (kosongkan)



Fungsi: Step ini akan membuang customer yang duplikat, hanya ambil pertama kali muncul

5. **Table Output – Menyimpan Data ke Dim_Customer**

Pada tahap ini, hasil proses Unique Rows akan dimuat ke tabel **Dim_Customer** pada database data warehouse.

Cara:

1. **Tambahkan Step Table Output**

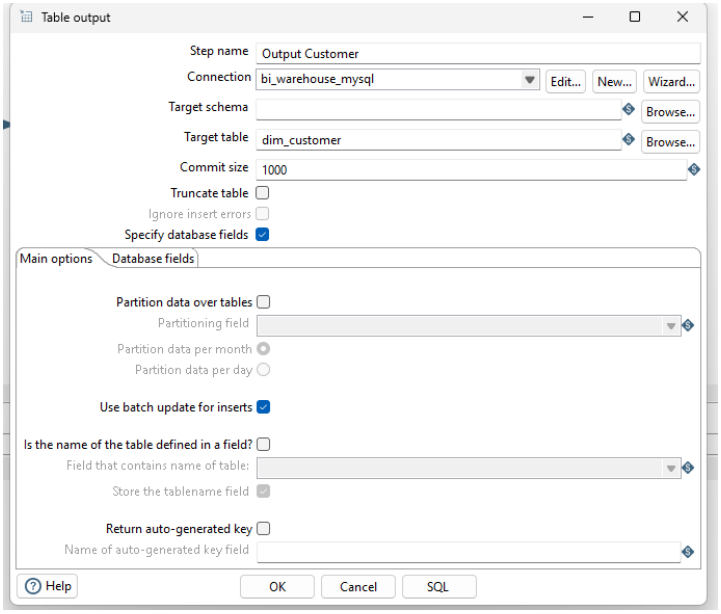
   Drag komponen **Table Output** dari panel *Output* ke workspace.

2. **Hubungkan Step**

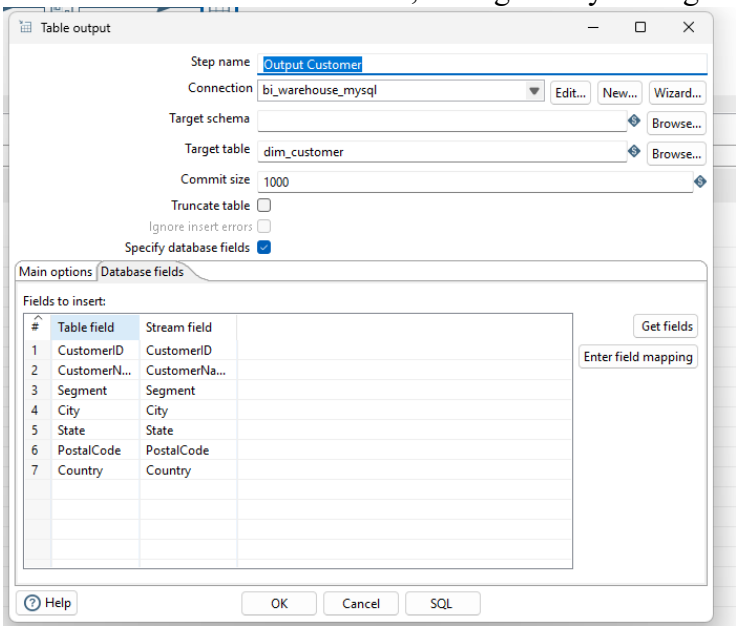
   Sambungkan *Unique Rows* ke *Table Output*.

3. **Konfigurasi Table Output**

1. Buka dengan double-click, lakukan konfigurasi seperti berikut

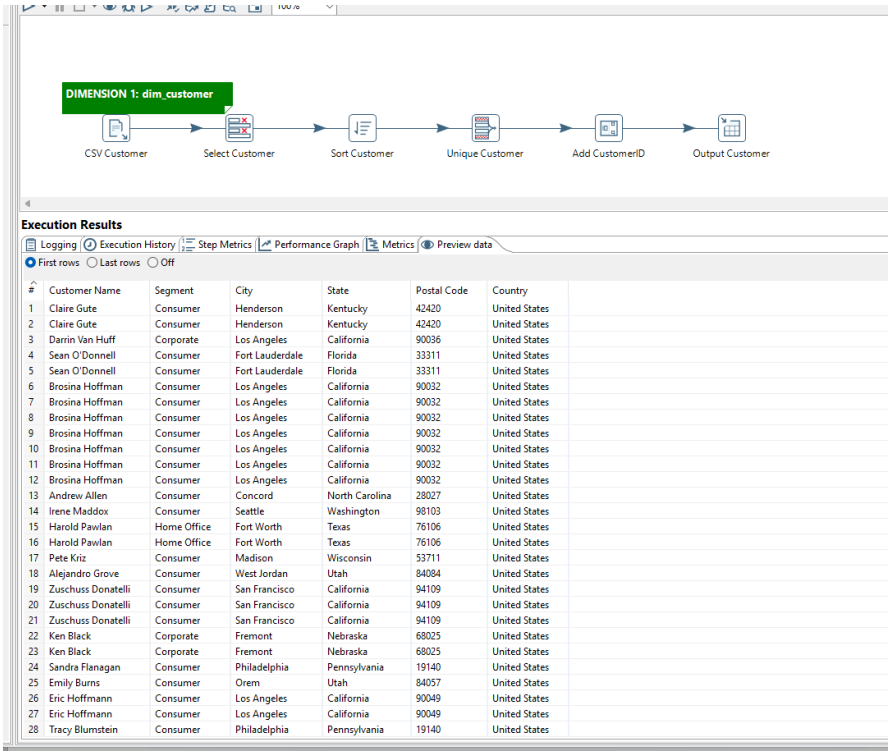


2. Pilih Connection: bi_warehouse_mysql
3. Target schema kosongkan saja
4. Target table: Dim_Customer
5. Centang:
    a. Specify database fields
    b. Use batch insert for inserts
6. Commit size: 1000, lalu OK
7. Pada tab Database Fields, konfigurasinya sebagai berikut



6. Preview & Execute

Setelah semua selesai lalu run dim_customer, lalu preview datanya, seperti berikut

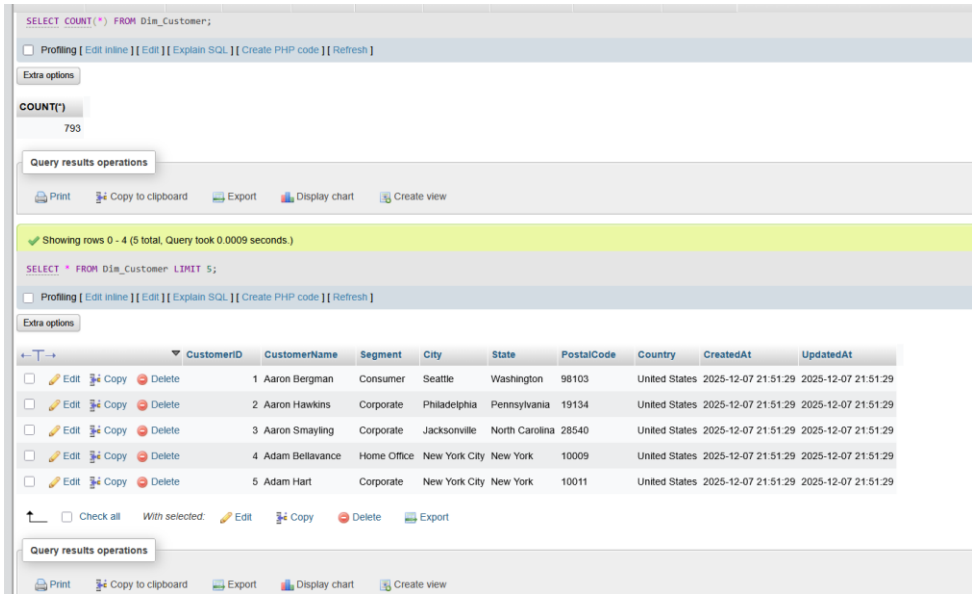


#	Customer Name	Segment	City	State	Postal Code	Country
1	Claire Gute	Consumer	Henderson	Kentucky	42420	United States
2	Claire Gute	Consumer	Henderson	Kentucky	42420	United States
3	Darrin Van Huff	Corporate	Los Angeles	California	90036	United States
4	Sean O'Donnell	Consumer	Fort Lauderdale	Florida	33311	United States
5	Sean O'Donnell	Consumer	Fort Lauderdale	Florida	33311	United States
6	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
7	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
8	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
9	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
10	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
11	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
12	Brosina Hoffman	Consumer	Los Angeles	California	90032	United States
13	Andrew Allen	Consumer	Concord	North Carolina	28027	United States
14	Irene Maddox	Consumer	Seattle	Washington	98103	United States
15	Harold Pawlan	Home Office	Fort Worth	Texas	76106	United States
16	Harold Pawlan	Home Office	Fort Worth	Texas	76106	United States
17	Pete Kriz	Consumer	Madison	Wisconsin	53711	United States
18	Alejandro Grove	Consumer	West Jordan	Utah	84084	United States
19	Zuschuss Donatelli	Consumer	San Francisco	California	94109	United States
20	Zuschuss Donatelli	Consumer	San Francisco	California	94109	United States
21	Zuschuss Donatelli	Consumer	San Francisco	California	94109	United States
22	Ken Black	Corporate	Fremont	Nebraska	68025	United States
23	Ken Black	Corporate	Fremont	Nebraska	68025	United States
24	Sandra Flanagan	Consumer	Philadelphia	Pennsylvania	19140	United States
25	Emily Burns	Consumer	Orem	Utah	84057	United States
26	Eric Hoffmann	Consumer	Los Angeles	California	90049	United States
27	Eric Hoffmann	Consumer	Los Angeles	California	90049	United States
28	Tracy Blumstein	Consumer	Philadelphia	Pennsylvania	19140	United States

Kita bisa cek pada step matrik untuk melihat jumlah cutomer

18	Output Customer	0	793	793	0	793	0	0	0	Finished	0.6s	1,365

Kita juga bisa melakukan pengeceakn di phpMyAdmin seperti berikut, dimana jumlahnya sama sama 793

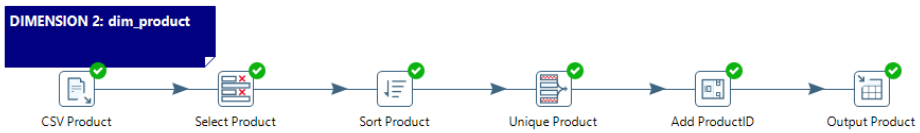


	CustomerID	CustomerName	Segment	City	State	PostalCode	Country	CreatedAt	UpdatedAt
Edit Copy Delete	1	Aaron Bergman	Consumer	Seattle	Washington	98103	United States	2025-12-07 21:51:29	2025-12-07 21:51:29
Edit Copy Delete	2	Aaron Hawkins	Corporate	Philadelphia	Pennsylvania	19134	United States	2025-12-07 21:51:29	2025-12-07 21:51:29
Edit Copy Delete	3	Aaron Smayling	Corporate	Jacksonville	North Carolina	28540	United States	2025-12-07 21:51:29	2025-12-07 21:51:29
Edit Copy Delete	4	Adam Bellavance	Home Office	New York City	New York	10009	United States	2025-12-07 21:51:29	2025-12-07 21:51:29
Edit Copy Delete	5	Adam Hart	Corporate	New York City	New York	10011	United States	2025-12-07 21:51:29	2025-12-07 21:51:29

1.4.3 Transformation 2: Load Dim_Produk

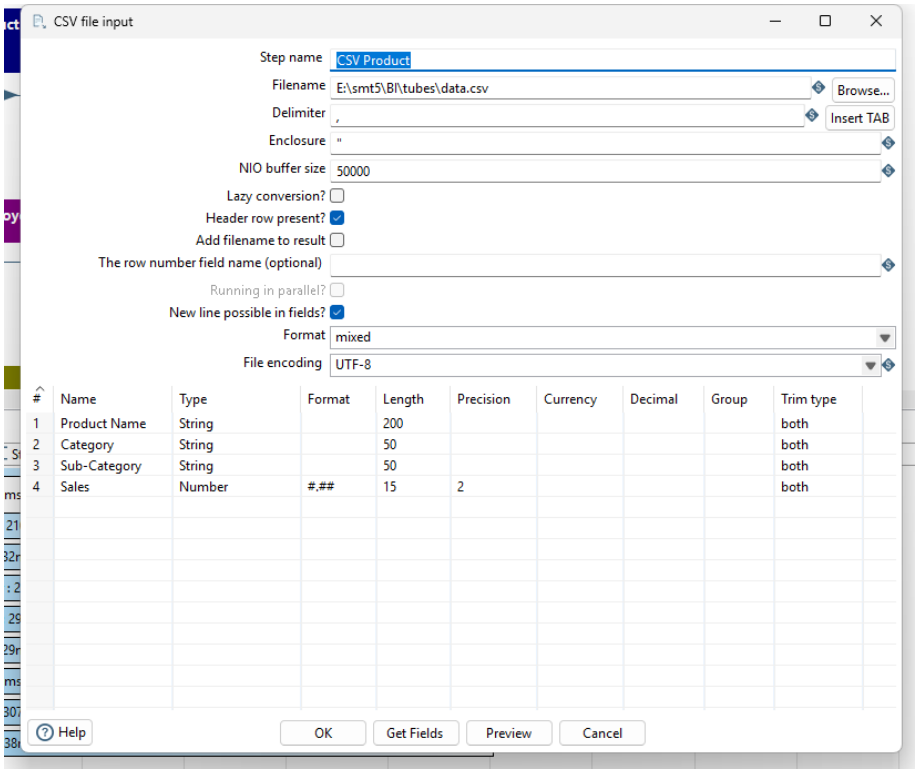
Transformasi **Load Dim_Product** bertujuan untuk mengekstrak data produk dari data.csv, membersihkan data, melakukan agregasi harga rata-rata berdasarkan penjualan per produk, menambahkan atribut manufacturer, dan memasukkannya ke tabel dimensional **Dim_Product**.

Transformasi ini juga memastikan bahwa setiap produk unik berdasarkan **Product ID**, dengan satu harga rata-rata (unit_price) yang dihitung secara otomatis

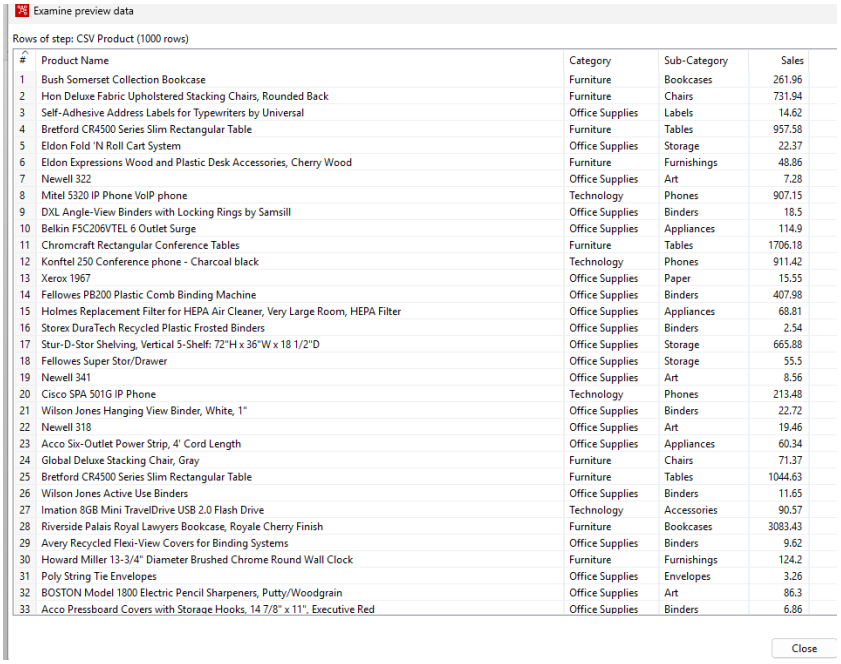


1. CSV Product

   Sama seperti pada Dim_Customer, cara penggunaannya adalah dengan drag step **CSV File Input** ke canvas, kemudian lakukan konfigurasi seperti berikut:



Lakukan preview jika perlu, seperti berikut, lalu close

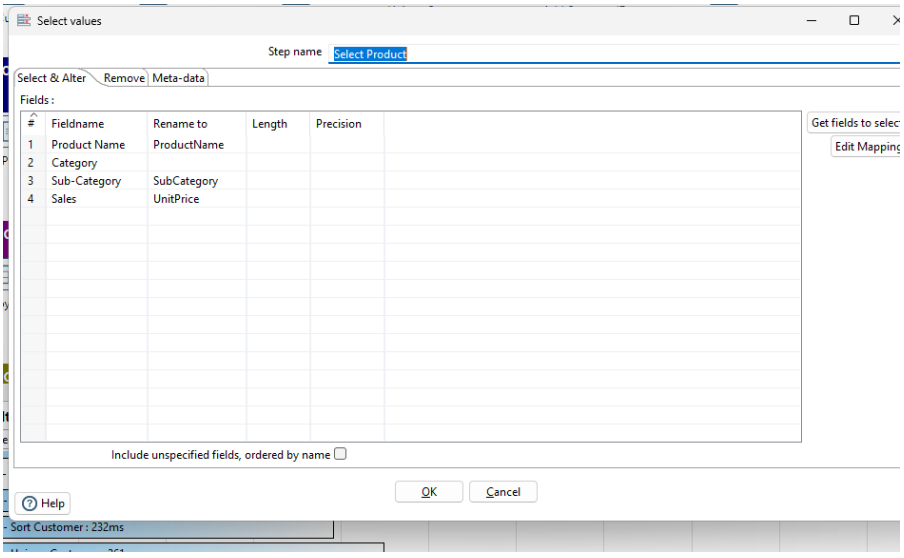


#	Product Name	Category	Sub-Category	Sales
1	Bush Somerset Collection Bookcase	Furniture	Bookcases	261.96
2	Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back	Furniture	Chairs	731.94
3	Self-Adhesive Address Labels for Typewriters by Universal	Office Supplies	Labels	14.62
4	Bretford CR4500 Series Slim Rectangular Table	Furniture	Tables	957.58
5	Eldon Fold 'N Roll Cart System	Office Supplies	Storage	22.37
6	Eldon Expressions Wood and Plastic Desk Accessories, Cherry Wood	Furniture	Furnishings	48.86
7	Newell 322	Office Supplies	Art	7.28
8	Mitel 5320 IP Phone VoIP phone	Technology	Phones	907.15
9	DXL Angle-View Binders with Locking Rings by Samsill	Office Supplies	Binders	18.5
10	Belkin F5C206VTEL 6 Outlet Surge	Office Supplies	Appliances	114.9
11	Chromcraft Rectangular Conference Tables	Furniture	Tables	1706.18
12	Konftel 250 Conference phone - Charcoal black	Technology	Phones	911.42
13	Xerox 1967	Office Supplies	Paper	15.55
14	Fellowes PB200 Plastic Comb Binding Machine	Office Supplies	Binders	407.98
15	Holmes Replacement Filter for HEPA Air Cleaner, Very Large Room, HEPA Filter	Office Supplies	Appliances	68.81
16	Storex DuraTech Recycled Plastic Frosted Binders	Office Supplies	Binders	2.54
17	Stur-D-Stor Shelving, Vertical 5-Shelf: 72"H x 36"W x 18 1/2"D	Office Supplies	Storage	665.88
18	Fellowes Super Stor/Drawer	Office Supplies	Storage	55.5
19	Newell 341	Office Supplies	Art	8.56
20	Cisco SPA 501G IP Phone	Technology	Phones	213.48
21	Wilson Jones Hanging View Binder, White, 1"	Office Supplies	Binders	22.72
22	Newell 318	Office Supplies	Art	19.46
23	Acco Six-Outlet Power Strip, 4' Cord Length	Office Supplies	Appliances	60.34
24	Global Deluxe Stacking Chair, Gray	Furniture	Chairs	71.37
25	Bretford CR4500 Series Slim Rectangular Table	Furniture	Tables	1044.63
26	Wilson Jones Active Use Binders	Office Supplies	Binders	11.65
27	Imation 8GB Mini TravelDrive USB 2.0 Flash Drive	Technology	Accessories	90.57
28	Riverside Palais Royal Lawyers Bookcase, Royale Cherry Finish	Furniture	Bookcases	3083.43
29	Avery Recycled Flexi-View Covers for Binding Systems	Office Supplies	Binders	9.62
30	Howard Miller 13-3/4" Diameter Brushed Chrome Round Wall Clock	Furniture	Furnishings	124.2
31	Poly String Tie Envelopes	Office Supplies	Envelopes	3.26
32	BOSTON Model 1800 Electric Pencil Sharpeners, Putty/Woodgrain	Office Supplies	Art	86.3
33	Acco Pressboard Covers with Storage Hooks, 14 7/8" x 11", Executive Red	Office Supplies	Binders	6.86

Jika sudah silahkan pilih OK,

2. Select Values

Kemudian, drag step **Select Values** ke canvas dan hubungkan dari **CSV File Input** ke **Select Values**, lalu klik dua kali, lalu lakukan konfigurasi seperti berikut



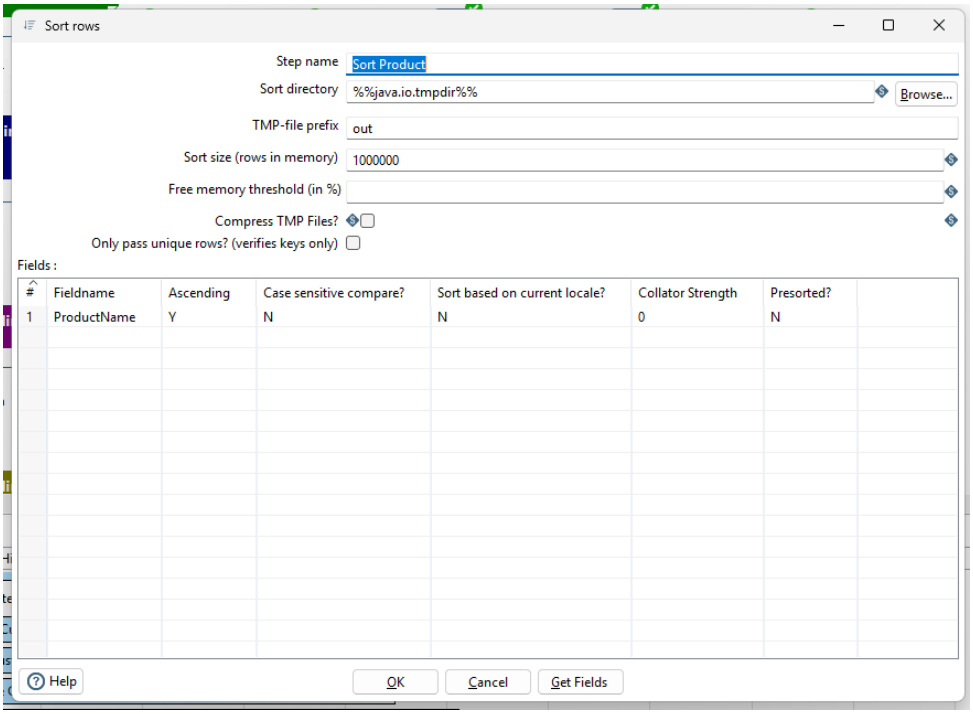
#	Fieldname	Rename to	Length	Precision
1	Product Name	ProductName		
2	Category			
3	Sub-Category	SubCategory		
4	Sales	UnitPrice		

3. Sort Rows

Pada step Sort Rows ini digunakan untuk mengurutkan data berdasarkan ProductName agar data lebih terstruktur sebelum dilakukan proses berikutnya. Drag step **Sort Rows** ke canvas, kemudian hubungkan dari

step sebelumnya. Setelah itu, lakukan konfigurasi pada bagian ***Fields*** seperti berikut

a. **Fieldname:** ProductName
b. **Ascending:** Y (Yes), diurutkan dari A ke Z / kecil ke besar
c. **Case sensitive compare:** N, tidak membedakan huruf besar/kecil
d. **Sort based on current locale:** N, tidak menggunakan locale setting
e. **Collator Strength:** 0, kekuatan perbandingan string
f. **Presorted:** N, data belum tersortir sebelumnya



4. Unique Rows

Unique rows di Pentaho Data Integration fungsinya untuk menghapus baris duplikat dari data stream. Drag step **Unique Rows** ke canvas, kemudian hubungkan dari step **Sort Rows**. Step ini harus diletakkan setelah Sort Rows karena data harus dalam keadaan terurut terlebih dahulu agar proses deteksi duplikat dapat bekerja dengan sesuai.

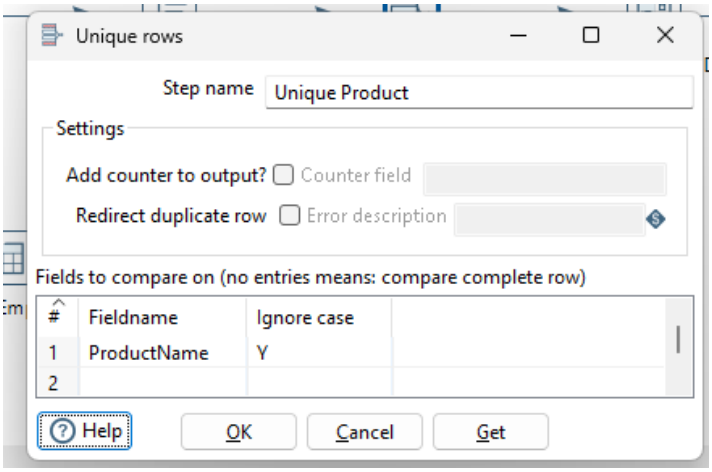
Konfigurasi :

Step name: Unique Product

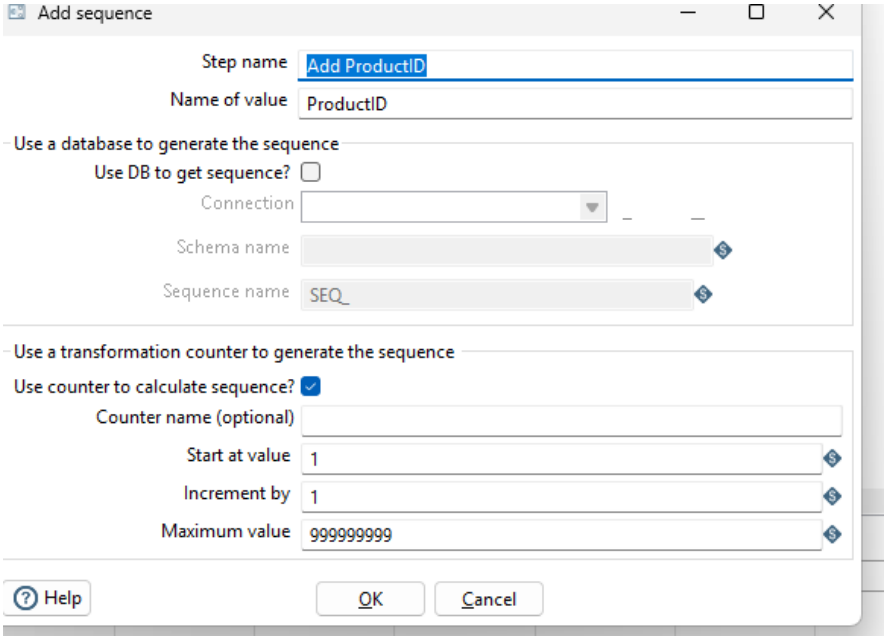
Pada **Add counter to output** tidak perlu dicentang, jika dicentang, akan menambahkan kolom counter yang menghitung jumlah baris duplikat yang ditemukan.

Redirect duplicate row, juga tidak perlu dicentang jika dicentang, baris duplikat akan diarahkan ke error handling stream, bukan dihapus.

Lalu isi , **Fieldname** isi ProductName dan **Ignore case:** Y (Yes), yaitu tidak membedakan huruf besar/kecil saat membandingkan



5. Add ProductID



6. Table Output

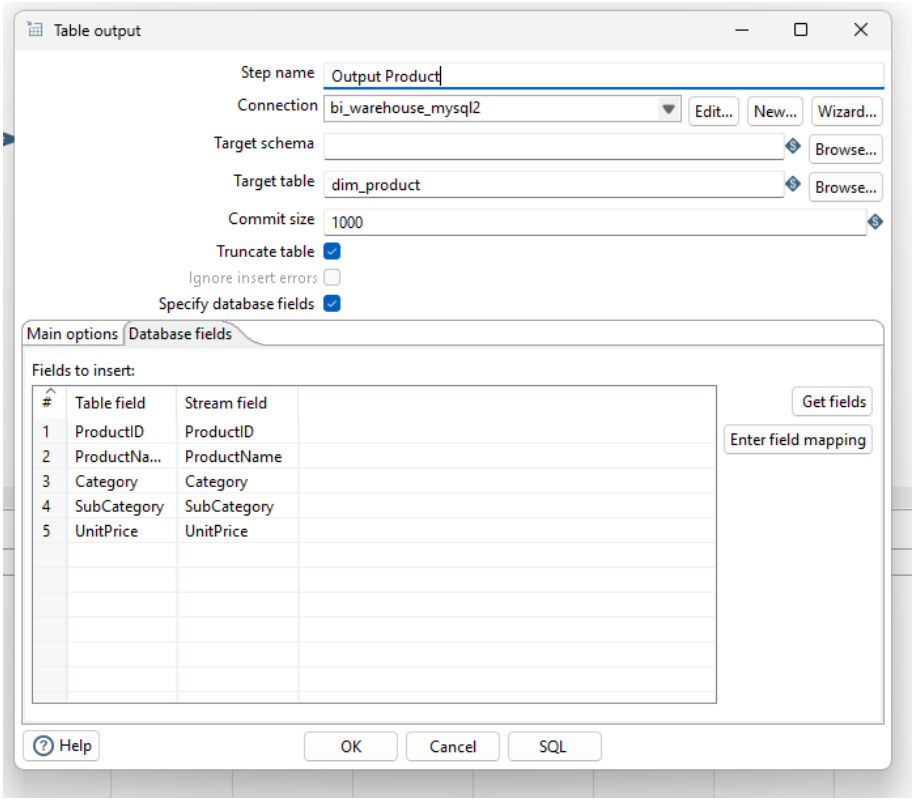
Step **Table Output** digunakan untuk **menyimpan hasil transformasi ETL ke dalam tabel di database Data Warehouse**.

Pada kasus ini, hasil pengolahan dimensi produk akan dimasukkan ke tabel **dim_product.** Step ini adalah **tahap terakhir** dari seluruh proses ETL Dim_Product, semua data yang sudah dibersihkan, diurutkan,

dihapus duplikatnya, dihitung rata-rata harganya, dan ditambah manufacturer, nanti disimpan ke database.

Cara:

1. Drag step **Table Output** ke canvas.
2. Hubungkan output dari step sebelumnya (misalnya Group By / Add Constants / Unique Rows) ke table output ini.
3. Buka pengaturan dan lakukan konfigurasi seperti pada gambar berikut



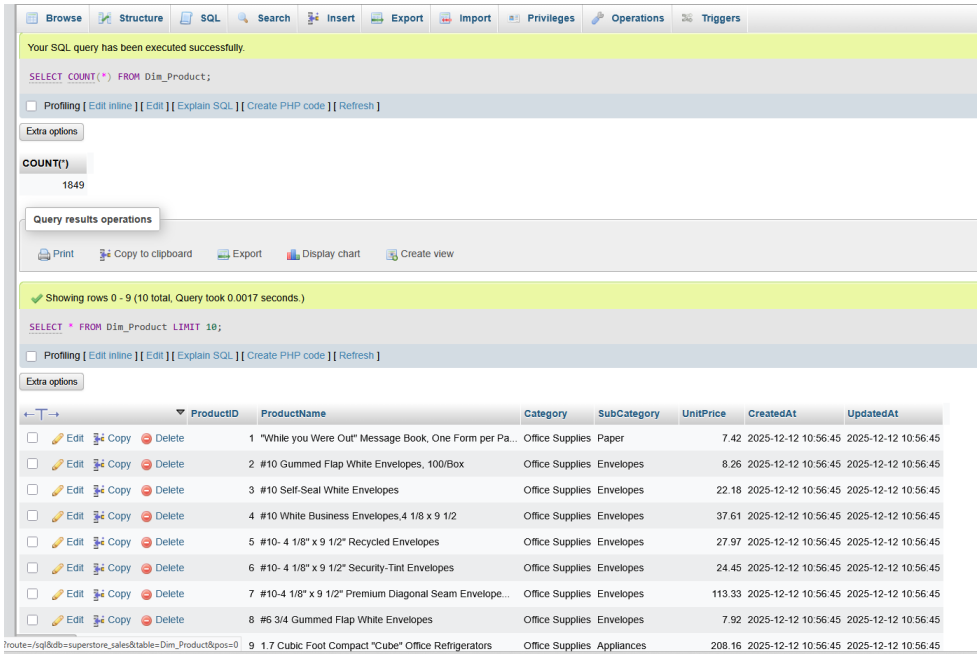
4. Jika sudah lalu klik OK

7. Table Output

Jika sudah kita *run*, kita lihat di *Step Metrics* bahwa jumlah datanya adalah **1849**.

13	Add ProductID	0	1849	1849	0	0	0	0	0	Finished	0.3s	5,438	-
14	Output Product	0	1849	1849	0	1849	0	0	0	Finished	0.8s	2,297	-
15	Output Employee	0	5	5	0	5	0	0	0	Finished	0.0s	111	-

Kita juga bisa cek pada phpMyAdmin untuk memastikan bahwa data yang masuk ke tabel tersebut memang sesuai.



Maka akan terlihat jumlah record yang sudah ter-insert, dan harus sama yaitu **1849 baris**. Jika jumlahnya cocok, berarti proses ETL untuk tabel dimensi ini berhasil berjalan dengan benar.

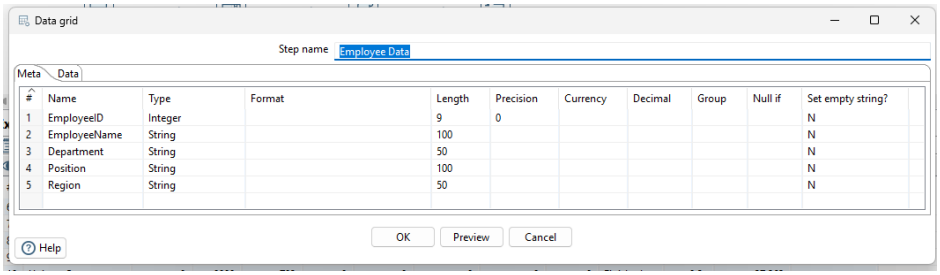
1.4.4 Transformation 3 : Load Dim_Employee

Pada transformasi ini dilakukan pembuatan tabel dimensi **Dim_Employee**. Karena pada file data.csv **tidak tersedia data karyawan**, maka data employee dibuat secara manual menggunakan fitur **Data Grid** pada Pentaho Data Integration. Data dummy ini digunakan untuk melengkapi struktur Data Warehouse dan mendukung analisis berbasis karyawan.

1. **Data Grid**

Cara:

1. Drag step **Data Grid** dari folder *Input* ke canvas.
2. Hubungkan langsung ke step selanjutnya (Table Output).
3. Double-click step **Data Grid** untuk melakukan konfigurasi field dan data seperti berikut



1. EmployeeID (Integer, 9)

   Field ini berfungsi sebagai primary key pada tabel dimensi employee serta memudahkan proses pengelompokan, pencarian, dan penghubungan data employee dengan tabel fakta di Data Warehouse

2. EmployeeName (String, 100)

   Digunakan untuk menyimpan nama lengkap karyawan secara , untuk kebutuhan pelaporan dan visualisasi data agar hasil analisis dapat dipahami oleh pengguna tanpa harus melihat kode atau ID karyawan.

3. Department (String, 50)

   Digunakan untuk menunjukkan departemen atau unit kerja tempat karyawan berada. Field ini memungkinkan analisis data berdasarkan struktur organisasi, seperti perbandingan kinerja antar departemen.

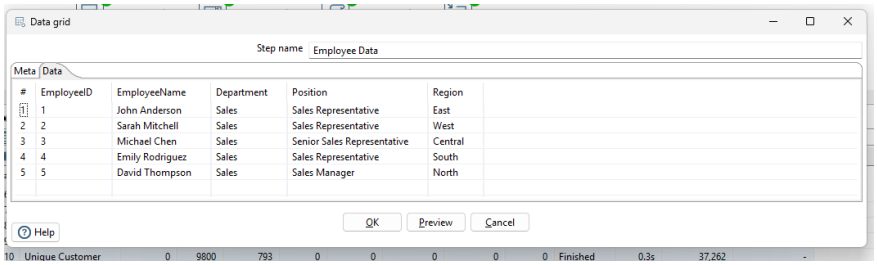
4. Position (String, 100)

   Digunakan untuk menyimpan informasi jabatan atau posisi karyawan dalam organisasi. Data ini dapat digunakan untuk analisis berdasarkan tingkat jabatan, tanggung jawab, atau peran karyawan.

5. Region (String, 50)

   Digunakan untuk menunjukkan wilayah atau area kerja karyawan. Informasi ini mendukung analisis distribusi karyawan dan performa berdasarkan lokasi atau wilayah kerja tertentu.

4. Pada tab Data, dilakukan pengisian data karyawan secara manual sesuai dengan struktur field yang telah ditentukan pada tab Meta. Data yang dimasukkan terdiri dari lima karyawan, sebagaimana ditunjukkan pada Tabel berikut:

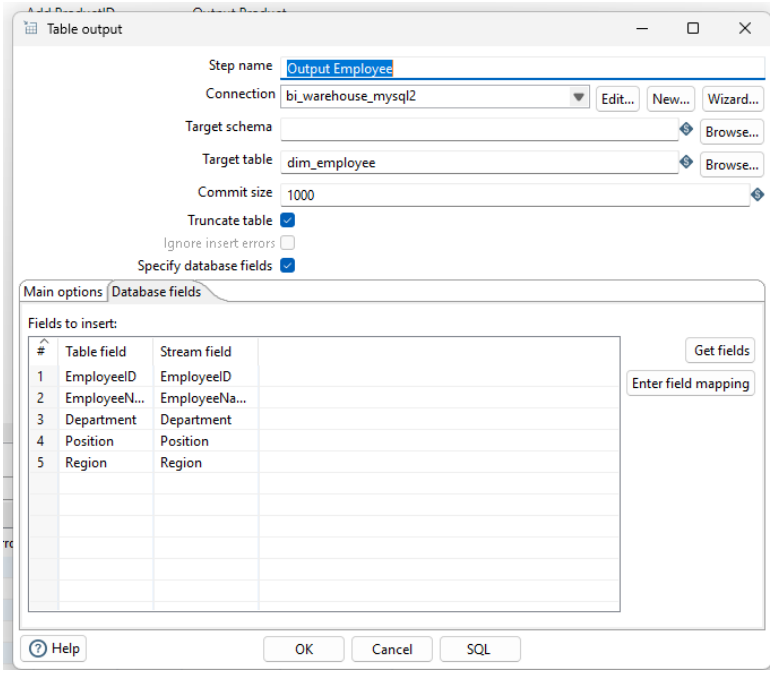


2. Table Output

Cara:

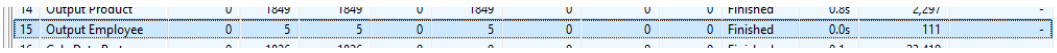
Setelah data karyawan berhasil dibuat pada step Data Grid, langkah selanjutnya adalah menyimpan data tersebut ke dalam tabel dimensi Dim_Employee. Untuk itu, drag step Table Output ke canvas dan hubungkan dengan step Data Grid.

Konfigurasi:



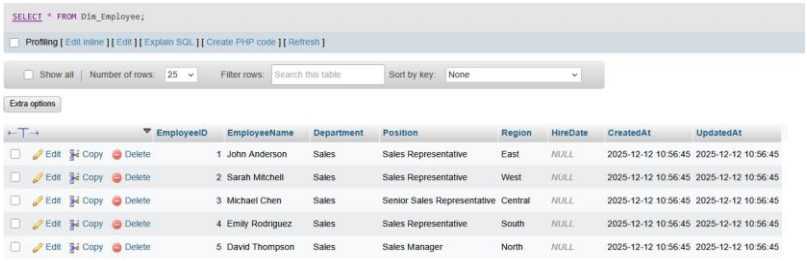
Verifikasi :

Output akan menghasilkan 5 data emloyee



Dan kita Kita juga bisa cek pada phpMyAdmin untuk memastikan bahwa data yang masuk ke tabel tersebut memang sesuai.

SELECT * FROM Dim_Employee;



1.4.5 Transformation 4 : Dim_Time

Transformation ini bertujuan untuk membangun **tabel dimensi waktu (Dim_Time)** yang digunakan sebagai referensi tanggal pada tabel fakta. Dimensi waktu sangat penting dalam data warehouse untuk mendukung analisis berbasis waktu seperti harian, bulanan, dan tahunan.

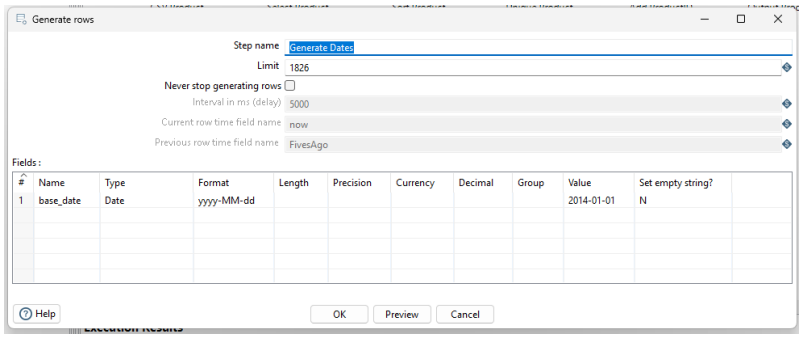
1. **Generate Dates**

   Step **Generate Dates** digunakan untuk menghasilkan data tanggal secara otomatis dalam rentang waktu tertentu tanpa bergantung pada dataset sumber.

   Cara:

   1. Drag step Generate Rows dari folder Input ke canvas transformasi
   2. Rename step menjadi Generate Dates,
   3. Double-click step tersebut untuk melakukan konfigurasi

   Konfigurasi :



   a. Step name: Generate Dates
   b. Limit: 1826, yaitu jumlah baris yang akan di-generate nilai ini setara dengan ± 5 tahun data tanggal (1826 hari 5 tahun)
   c. Never stop generating rows tidak dicentang Proses akan berhenti setelah jumlah baris mencapai nilai limit

d. Isi fields dengan Name **base_date**, adalah kolom tanggal yang akan di-generate

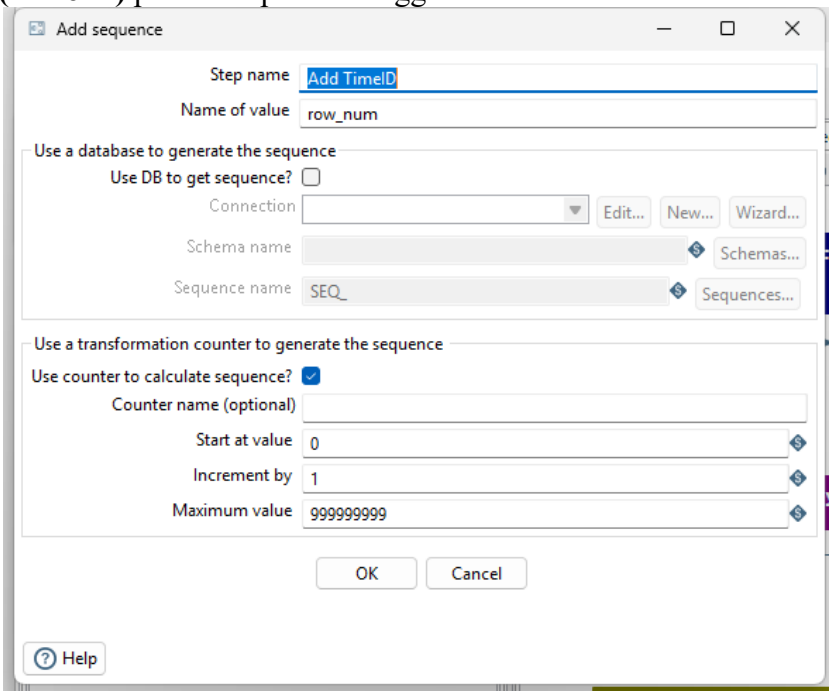
e. Tanggal dimulai dari **1 Januari 2014**

Setiap baris akan menghasilkan tanggal berurutan missal 2014-01-01, 2014-01-02, 2014-01-03 dan seterusnya hingga 1826 baris

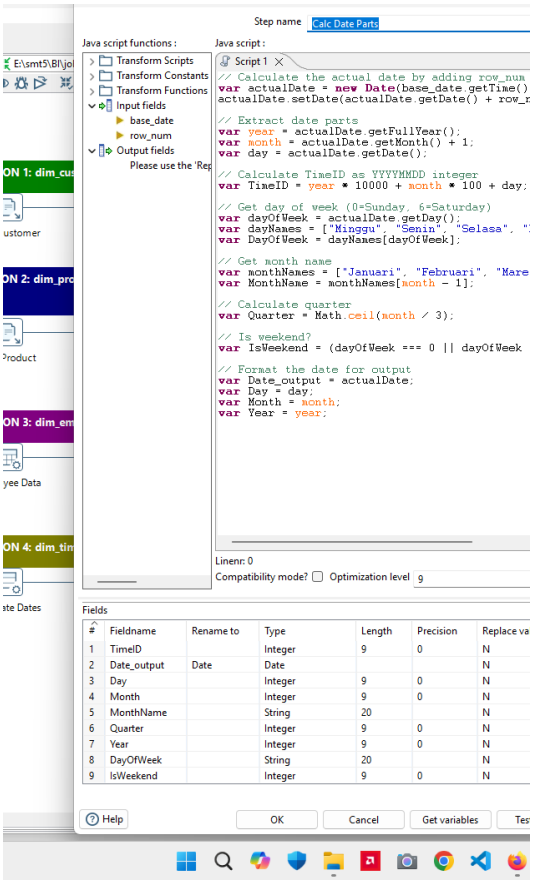
f. Step ini menghasilkan dataset tanggal berurutan yang akan digunakan sebagai dasar pembentukan dimensi waktu (Dim_Time) sebelum ditambahkan TimeID dan atribut tanggal lainnya.

2. **Add Sequence**

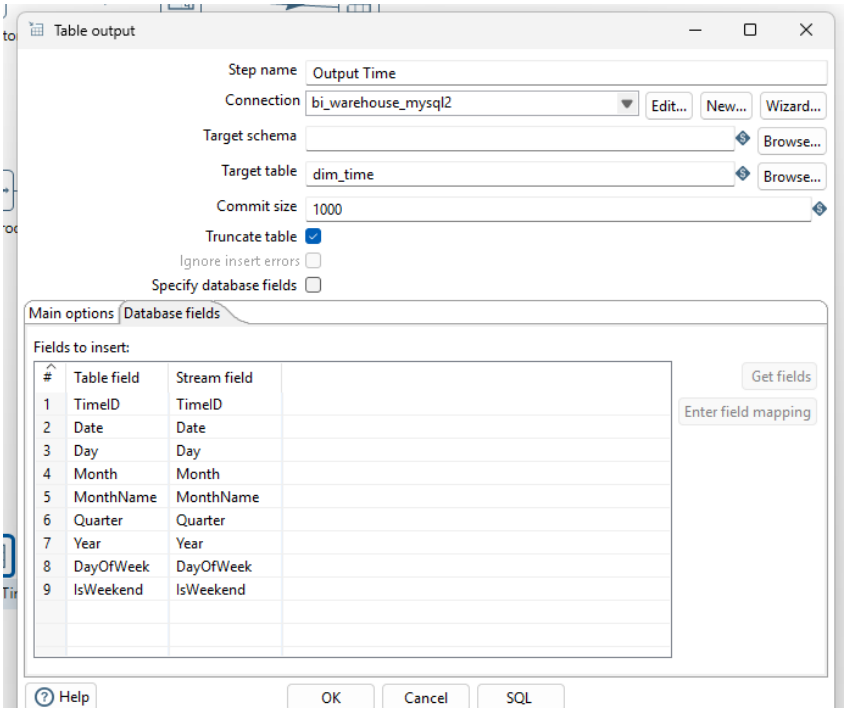
Step **Add TimeID** digunakan untuk menambahkan **primary key waktu (TimeID)** pada setiap baris tanggal.



3. Calc Date Parts

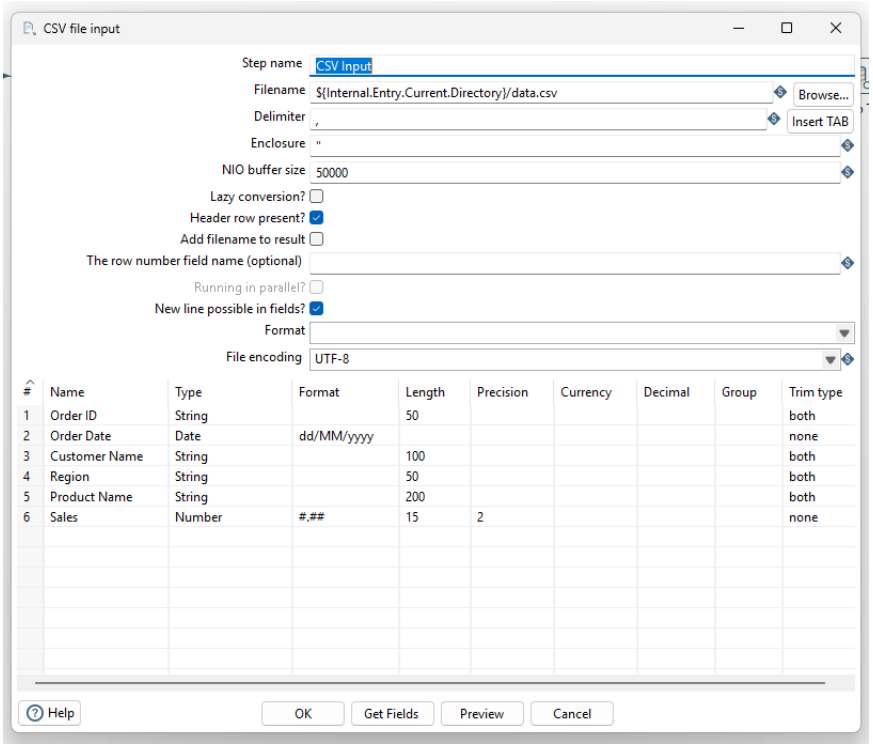


4. Tabel Output

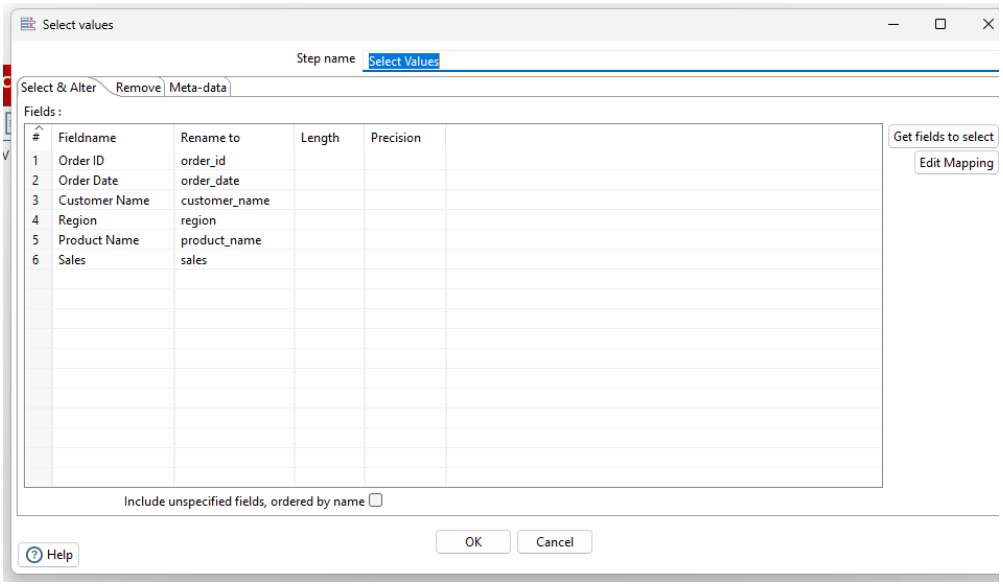


1.4.6 Transformasi Fakta

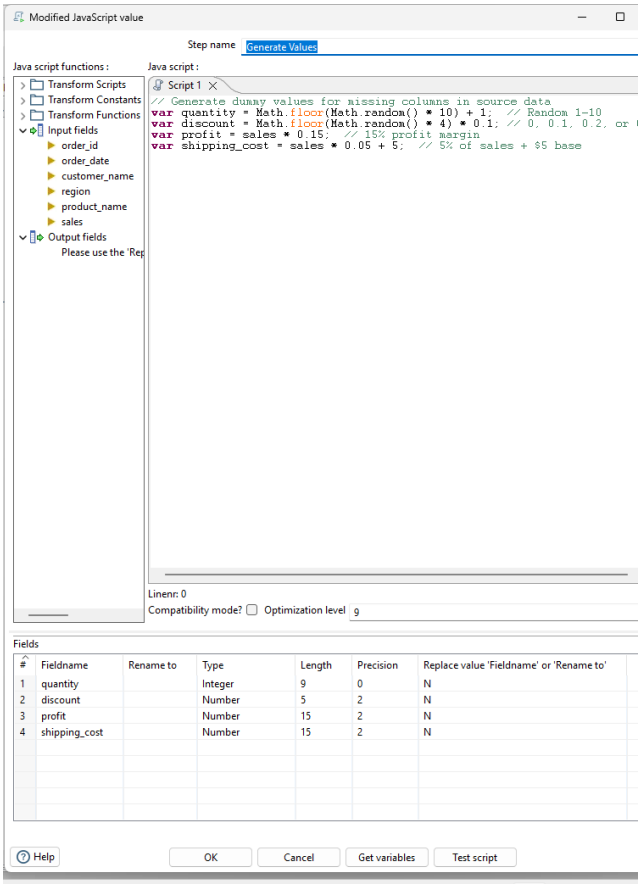
1. CSV file Input



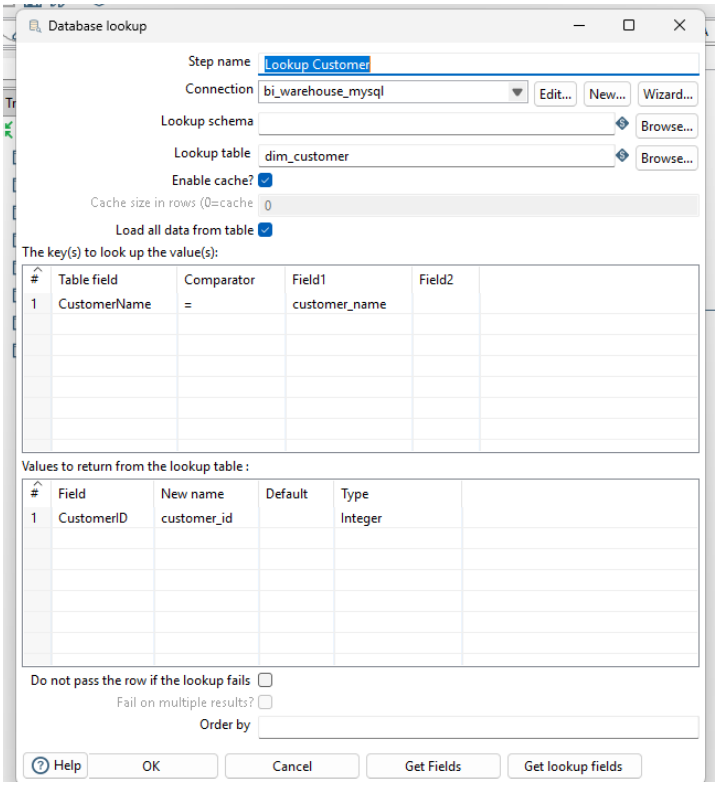
2. Select Values



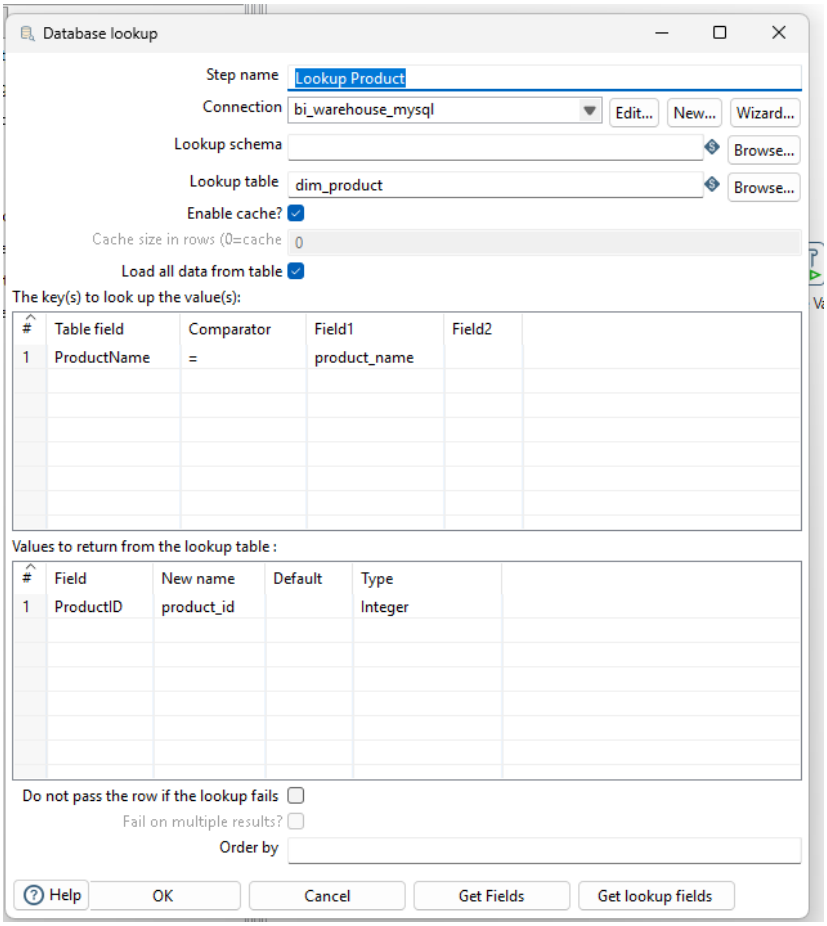
3. Generate Values



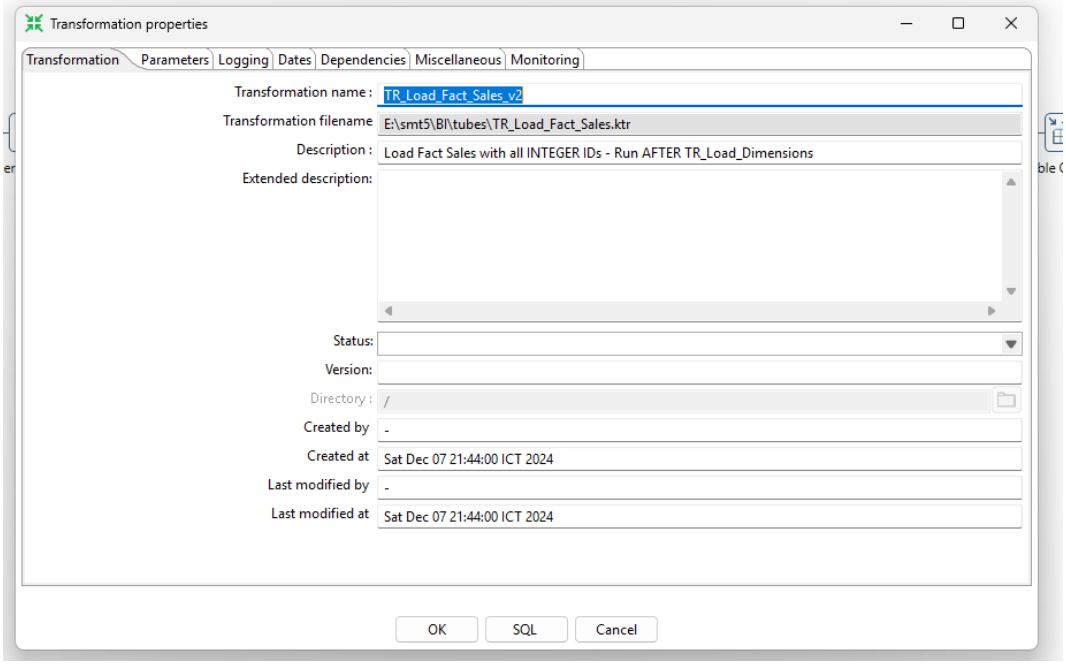
4. Lookup Customer



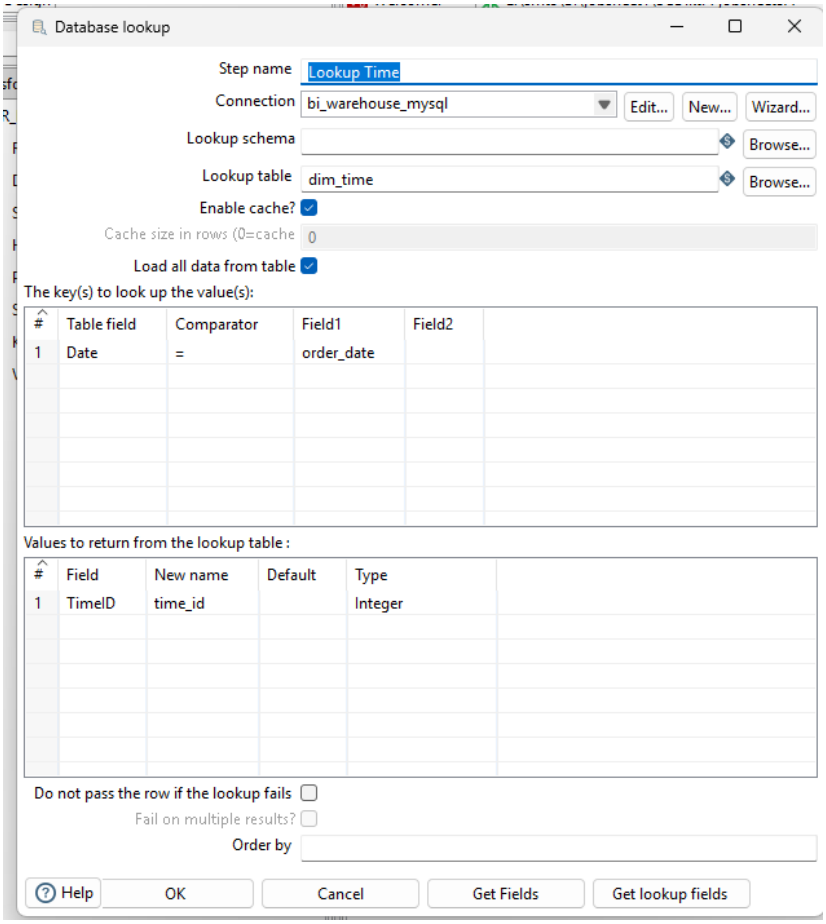
5. Lookup Product



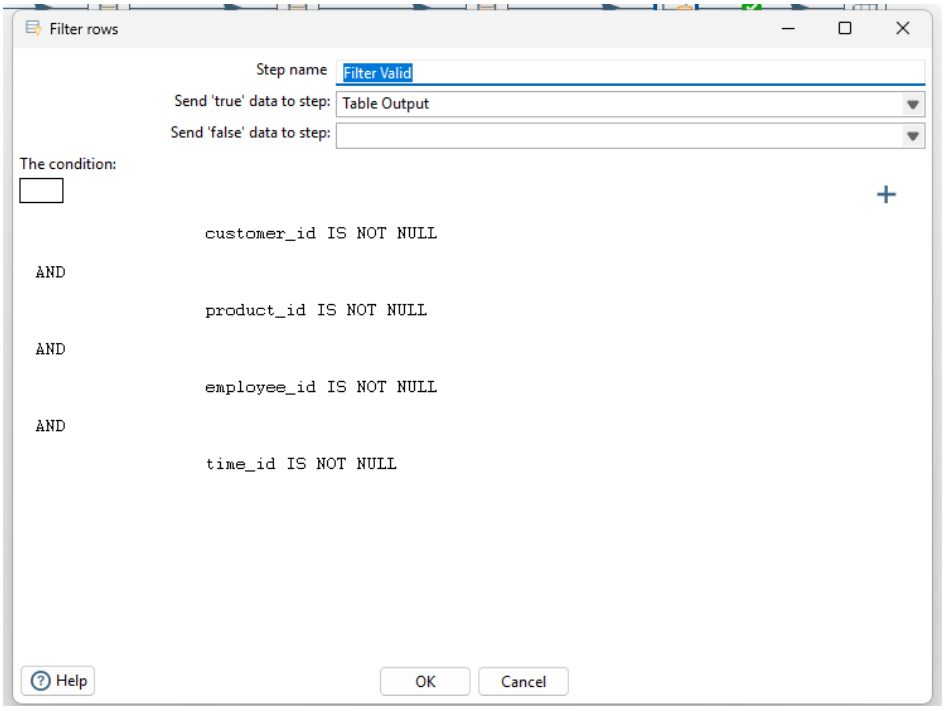
6. Lookup Employee



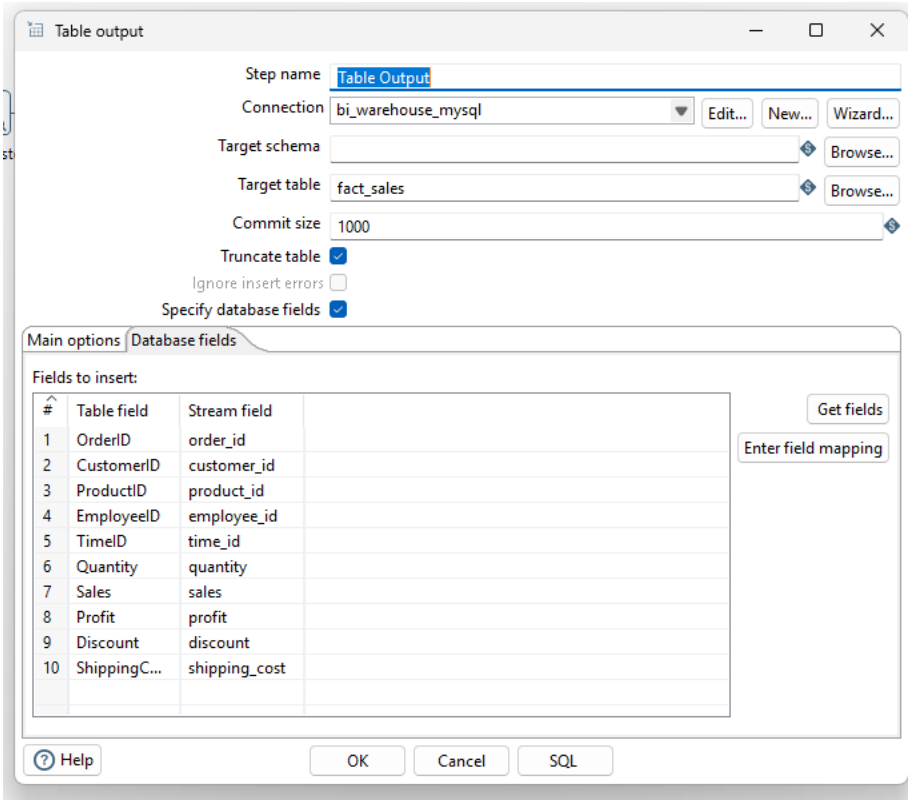
7. Lookup Time



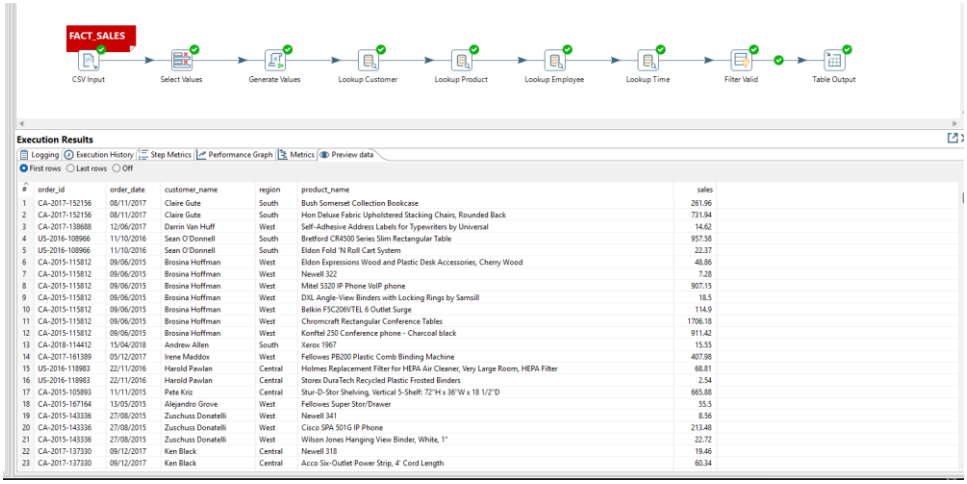
8. Filter Valid



9. Table Output



10. Hasil saat semua dihubungkan dan dirun

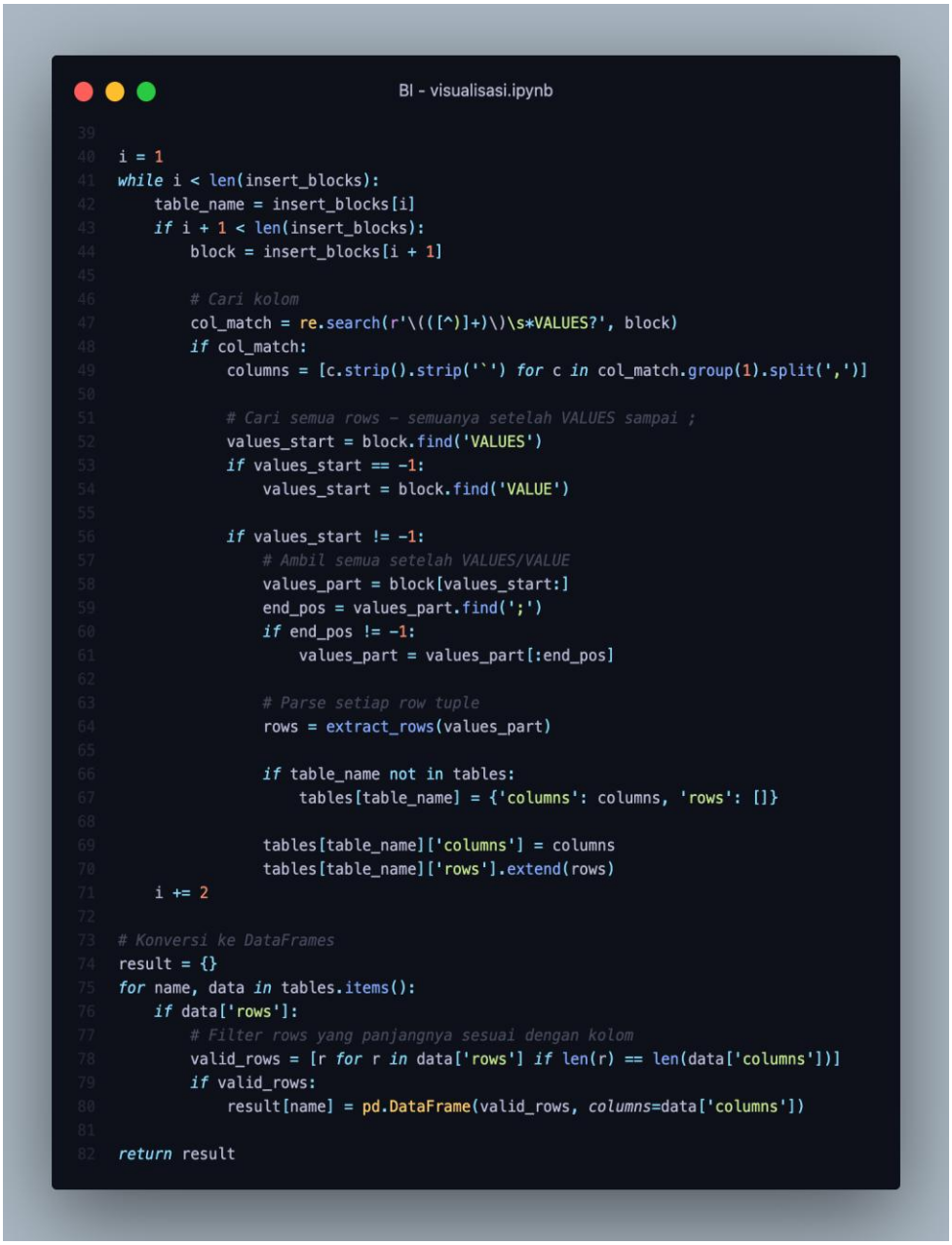


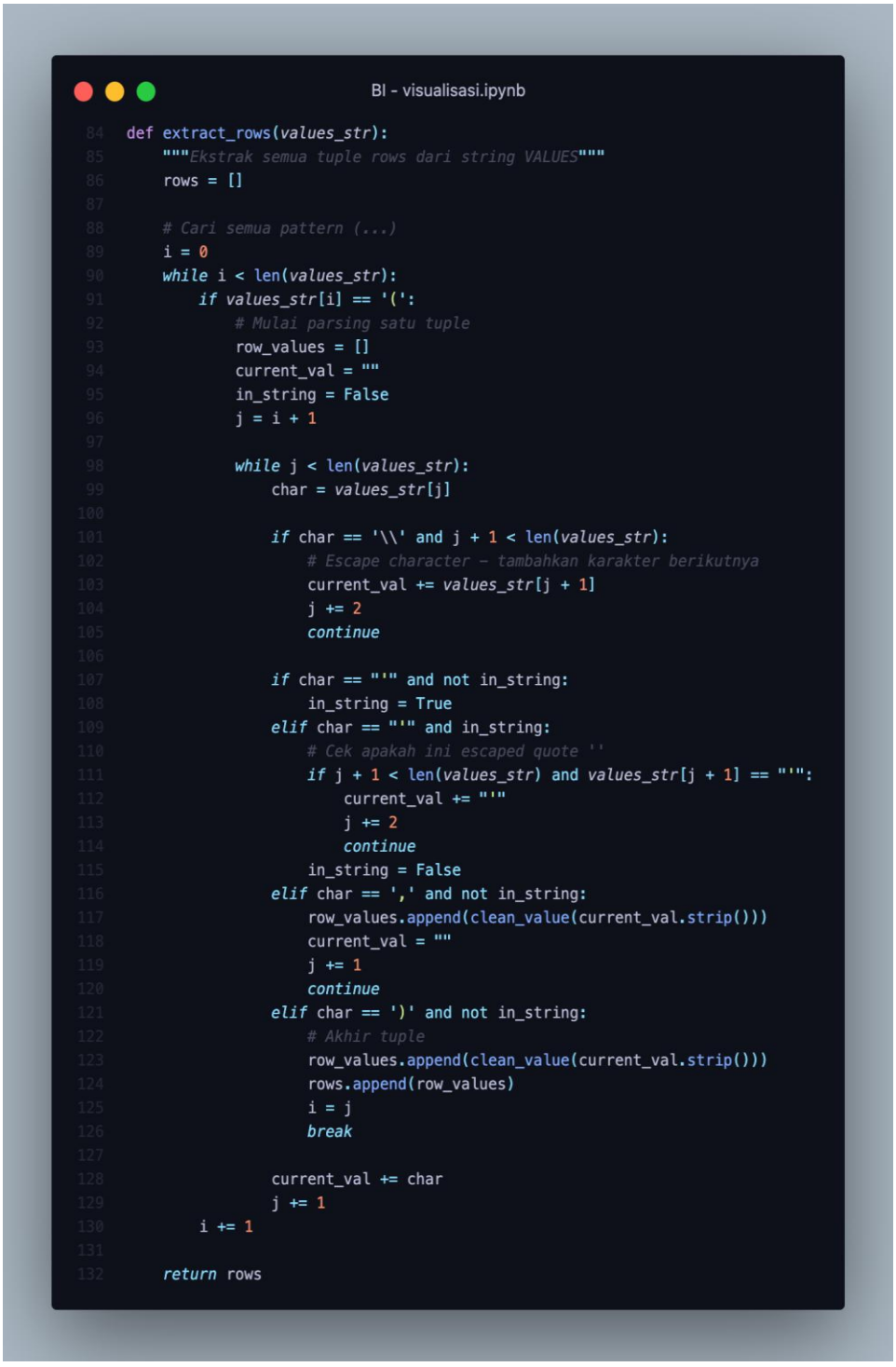
1.5 VISUALISASI

Setelah proses ETL selesai dan data berhasil dimuat ke dalam data warehouse, tahap berikutnya adalah visualisasi data untuk menyajikan hasil analisis kinerja penjualan secara grafis. Pada tahapan ini, visualisasi dilakukan dengan menggunakan bahasa pemrograman Python. Pemilihan tools ini didasarkan pada volume data yang diolah cukup besar, sehingga penggunaan tools Business Intelligence (BI) versi gratis seperti Power BI atau Google Looker dinilai kurang efisien dari sisi performa dan membutuhkan waktu pemrosesan (rendering time) yang cukup lama.

Berikut adalah langkah untuk melakukan visualisasi dengan menggunakan python:

1. Menghubungkan dengan database lalu parsing menjadi sebuah dataframe yang siap olah


BI - visualisasi.ipynb
i = 1
while i < len(insert_blocks):
table_name = insert_blocks[i]
if i + 1 < len(insert_blocks):
block = insert_blocks[i + 1]
# Cari kolom
col_match = re.search(r'\(([^)]+)\)\s*VALUES?', block)
if col_match:
columns = [c.strip().strip('`') for c in col_match.group(1).split(',')]
# Cari semua rows – semuanya setelah VALUES sampai ;
values_start = block.find('VALUES')
if values_start == -1:
values_start = block.find('VALUE')
if values_start != -1:
# Ambil semua setelah VALUES/VALUE
values_part = block[values_start:]
end_pos = values_part.find(';')
if end_pos != -1:
values_part = values_part[:end_pos]
# Parse setiap row tuple
rows = extract_rows(values_part)
if table_name not in tables:
tables[table_name] = {'columns': columns, 'rows': []}
tables[table_name]['columns'] = columns
tables[table_name]['rows'].extend(rows)
i += 2
# Konversi ke DataFrames
result = {}
for name, data in tables.items():
if data['rows']:
# Filter rows yang panjangnya sesuai dengan kolom
valid_rows = [r for r in data['rows'] if len(r) == len(data['columns'])]
if valid_rows:
result[name] = pd.DataFrame(valid_rows, columns=data['columns'])
return result


BI - visualisasi.ipynb
def extract_rows(values_str):
"""Ekstrak semua tuple rows dari string VALUES"""
rows = []
# Cari semua pattern (...)
i = 0
while i < len(values_str):
if values_str[i] == '(':
# Mulai parsing satu tuple
row_values = []
current_val = ""
in_string = False
j = i + 1
while j < len(values_str):
char = values_str[j]
if char == '\\' and j + 1 < len(values_str):
# Escape character - tambahkan karakter berikutnya
current_val += values_str[j + 1]
j += 2
continue
if char == "'" and not in_string:
in_string = True
elif char == "'" and in_string:
# Cek apakah ini escaped quote ''
if j + 1 < len(values_str) and values_str[j + 1] == "'":
current_val += "'"
j += 2
continue
in_string = False
elif char == ',' and not in_string:
row_values.append(clean_value(current_val.strip()))
current_val = ""
j += 1
continue
elif char == ')' and not in_string:
# Akhir tuple
row_values.append(clean_value(current_val.strip()))
rows.append(row_values)
i = j
break
current_val += char
j += 1
i += 1
return rows

BI - visualisasi.ipynb

```python
def clean_value(val):
    """Bersihkan dan konversi nilai dari SQL"""
    if val.upper() == 'NULL':
        return None
    if val.startswith("'") and val.endswith("'"):
        return val[1:-1]
    try:
        if '.' in val:
            return float(val)
        return int(val)
    except:
        return val

# 3. Parse file SQL
print("Memproses file SQL...")
tables = parse_mysql_dump_simple(sql_file_path)

print("\nTabel yang ditemukan:")
for name, tbl in tables.items():
    print(f"  - {name}: {len(tbl)} baris, kolom: {list(tbl.columns)}")

# 4. Gabungkan tabel-tabel
fact_sales = tables.get('fact_sales', pd.DataFrame())
dim_time = tables.get('dim_time', pd.DataFrame())
dim_product = tables.get('dim_product', pd.DataFrame())
dim_customer = tables.get('dim_customer', pd.DataFrame())
dim_employee = tables.get('dim_employee', pd.DataFrame())

# Merge semua tabel dengan pengecekan
df = fact_sales.copy()

if not df.empty:
    if not dim_time.empty and 'TimeID' in df.columns and 'TimeID' in dim_time.columns:
        df = df.merge(dim_time, on='TimeID', how='left', suffixes=('', '_time'))
    if not dim_product.empty and 'ProductID' in df.columns and 'ProductID' in dim_product.columns:
        df = df.merge(dim_product, on='ProductID', how='left', suffixes=('', '_product'))
    if not dim_customer.empty and 'CustomerID' in df.columns and 'CustomerID' in dim_customer.columns:
        df = df.merge(dim_customer, on='CustomerID', how='left', suffixes=('', '_customer'))
    if not dim_employee.empty and 'EmployeeID' in df.columns and 'EmployeeID' in dim_employee.columns:
        df = df.merge(dim_employee, on='EmployeeID', how='left', suffixes=('', '_employee'))

# Konversi kolom Date ke datetime
if 'Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
    df['YearMonth'] = df['Date'].dt.to_period('M')

print("\nData berhasil dimuat. Jumlah baris:", len(df))
print("\nKolom tersedia:", list(df.columns))
print("\nPreview data:")
print(df.head())
```

2. Membuat visualisasi untuk masing-masing KPI
    a. Tren Bulanan

    b. Produk dengan pendapatan terbesar

    c. Total penjualan berdasarkan segmen customer

    d. Performa penjualan antar wilayah

BI - visualisasi.ipynb

```python
# Agregasi berdasarkan Region
region_sales = df.groupby('Region')['Sales'].sum().reset_index()

plt.figure(figsize=(10, 6))
colors = ['#1f77b4', '#d69e2e', '#0f9d58', '#c0504d']
sns.barplot(data=region_sales, x='Region', y='Sales', palette=colors)

plt.title('Total Sales by Region', fontsize=16)
plt.xlabel('Region', fontsize=12)
plt.ylabel('Sales', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

e. Margin keuntungan tiap kategori produk

BI - visualisasi.ipynb

```python
# Agregasi berdasarkan Category untuk Profit
category_profit = df.groupby('Category')['Profit'].sum().reset_index()

plt.figure(figsize=(10, 6))
colors = ['#1f77b4', '#d69e2e', '#0f9d58']
sns.barplot(data=category_profit, x='Category', y='Profit', palette=colors)

plt.title('Total Profit by Category', fontsize=16)
plt.xlabel('Category', fontsize=12)
plt.ylabel('Profit', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

Setelah selesai dalam visualisasi, berikut saya akan menjelaskan hasil diagram yang telah divisualisasikan sebelumnya:

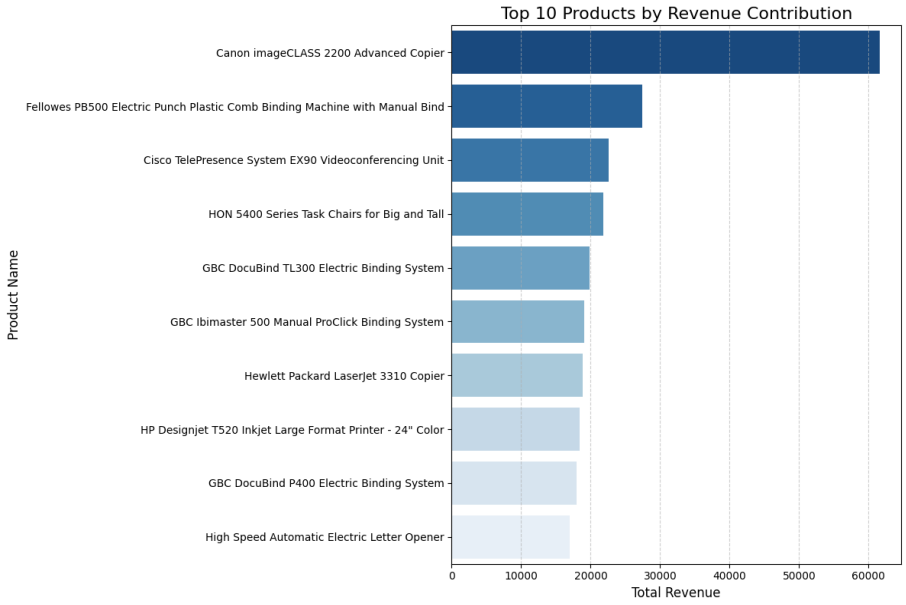
1. Tren bulanan untuk melihat performa penjualan dan profit.

Grafik garis ini menggambarkan total pendapatan penjualan (sales) yang dikelompokkan berdasarkan bulan dan tahun (YearMonth).

Berdasarkan visualisasi tersebut, terlihat adanya fluktuasi penjualan yang dinamis sepanjang periode.

Puncak penjualan (peak sales) cenderung terjadi pada bulan-bulan akhir tahun (seperti November dan Desember), yang mengindikasikan adanya pola musiman (seasonality) di mana permintaan pelanggan meningkat, kemungkinan besar dipicu oleh musim liburan atau diskon akhir tahun. Sebaliknya, terdapat penurunan signifikan di awal tahun. Informasi ini penting bagi manajemen untuk merencanakan strategi inventaris agar stok tersedia saat permintaan tinggi dan strategi promosi saat permintaan rendah.

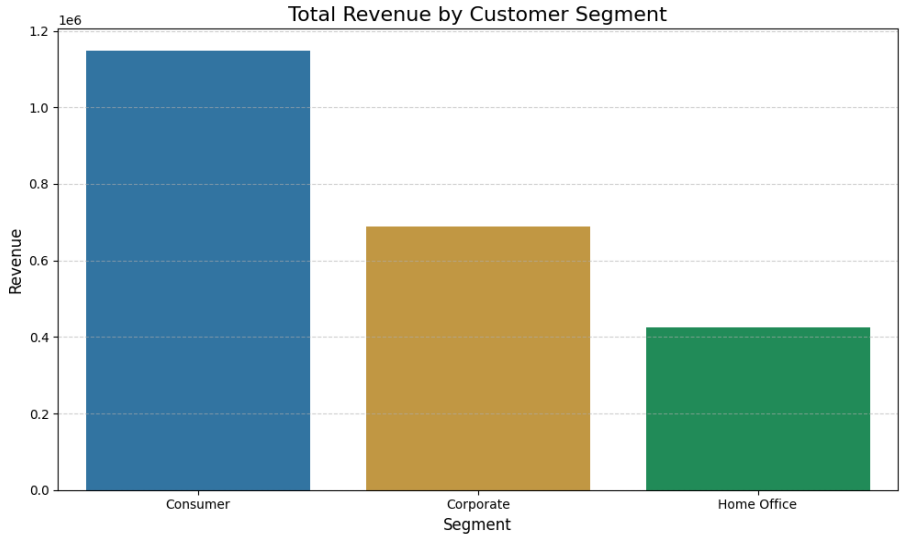
2. Produk dengan kontribusi pendapatan terbesar.



Grafik batang horizontal ini menampilkan 10 produk dengan total nilai penjualan tertinggi. Produk seperti Canon imageCLASS dan mesin penjilidan (Binding Machine) terlihat mendominasi pendapatan dibandingkan ribuan produk lainnya.

Analisis ini menunjukkan penerapan Prinsip Pareto (Aturan 80/20), di mana sebagian kecil produk menghasilkan porsi pendapatan yang besar. Bagi perusahaan, produk-produk dalam daftar 'Top 10' ini adalah aset vital yang ketersediaannya harus selalu dijaga (jangan sampai out of stock) dan dapat menjadi fokus utama dalam kampanye pemasaran premium.

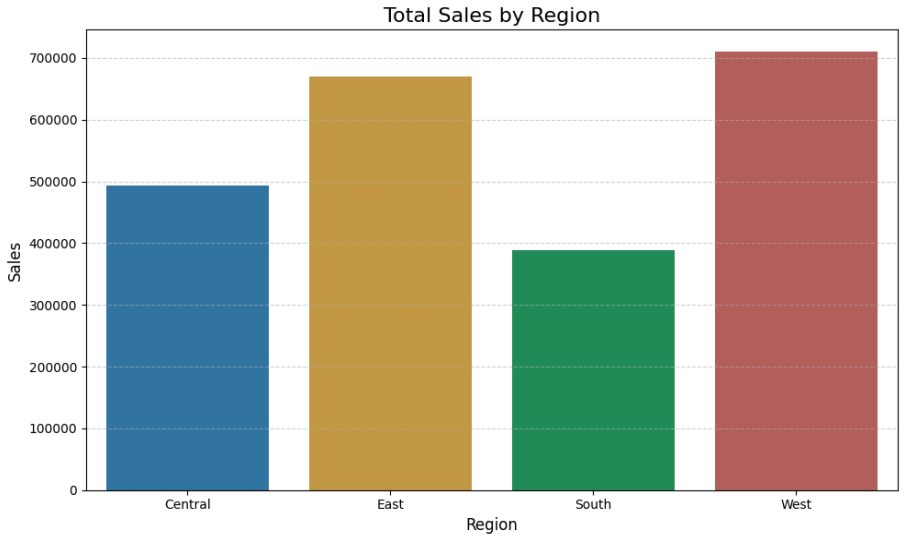
3. Proporsi penjualan berdasarkan segmen pelanggan.



Visualisasi ini membandingkan total pendapatan yang dihasilkan dari tiga segmen pelanggan utama. Terlihat jelas bahwa segmen Consumer (Konsumen Perorangan) memberikan kontribusi pendapatan terbesar, yang jauh melampaui segmen Corporate dan Home Office.

Hal ini mengindikasikan bahwa fokus bisnis Global Superstore saat ini lebih kuat di pasar Business-to-Consumer (B2C). Sebagai rekomendasi strategis, perusahaan dapat mempertahankan loyalitas segmen Consumer dengan program member, sembari mencari peluang untuk meningkatkan penetrasi pasar di sektor Corporate yang masih memiliki potensi pertumbuhan.

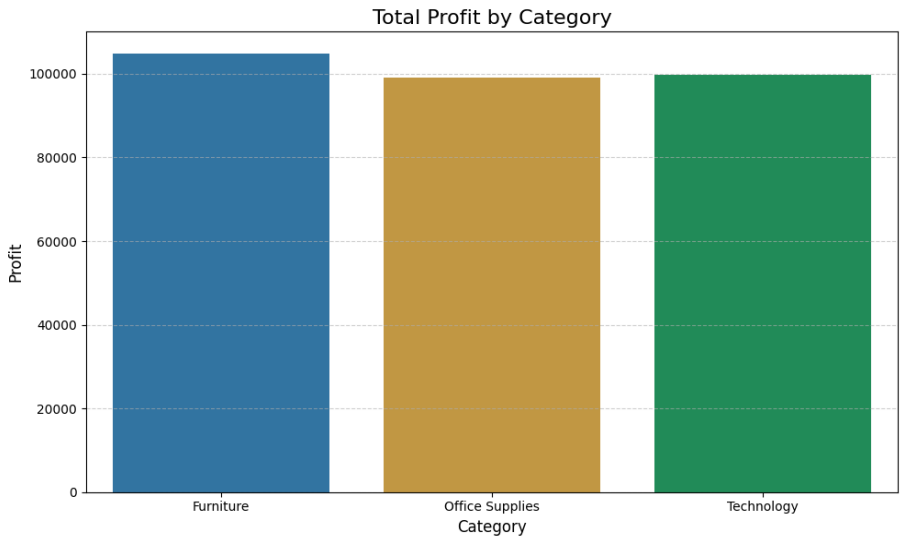
4. Perbandingan performa penjualan antar wilayah.



Grafik ini memetakan total penjualan berdasarkan empat wilayah operasional: South, West, Central, dan East. Dari visualisasi, terlihat bahwa wilayah West (Barat) dan East (Timur) adalah penyumbang penjualan terbesar, sementara wilayah South (Selatan) memiliki kinerja yang paling rendah.

Ketimpangan ini menunjukkan bahwa basis pelanggan perusahaan terkonsentrasi di pesisir Barat dan Timur. Manajemen perlu melakukan evaluasi mendalam terhadap wilayah Selatan; apakah rendahnya penjualan disebabkan oleh kurangnya pemasaran, biaya pengiriman yang tinggi, atau persaingan lokal yang ketat, sehingga strategi ekspansi yang tepat dapat dirumuskan.

5. Analisis margin keuntungan per kategori produk.



Berbeda dengan grafik penjualan (*sales*), grafik ini berfokus pada Profit (Keuntungan Bersih) yang dihasilkan oleh kategori *Furniture, Office Supplies,* dan *Technology*.

Analisis ini krusial karena produk dengan penjualan tinggi belum tentu menghasilkan profit yang besar jika biaya operasional atau diskonnya tinggi. Visualisasi ini membantu manajemen menilai efektivitas harga dan biaya. Jika kategori *Technology* memiliki profit tertinggi, maka perusahaan dapat mendorong penjualan produk teknologi bernilai tinggi. Sebaliknya, jika *Furniture* memiliki profit rendah meskipun penjualannya tinggi,

mungkin perlu dilakukan peninjauan ulang terhadap biaya logistik/pengiriman untuk kategori tersebut.

Link Github : https://github.com/febriansyahadin/Business-Intelligence.git