

**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/rohithsahoo/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

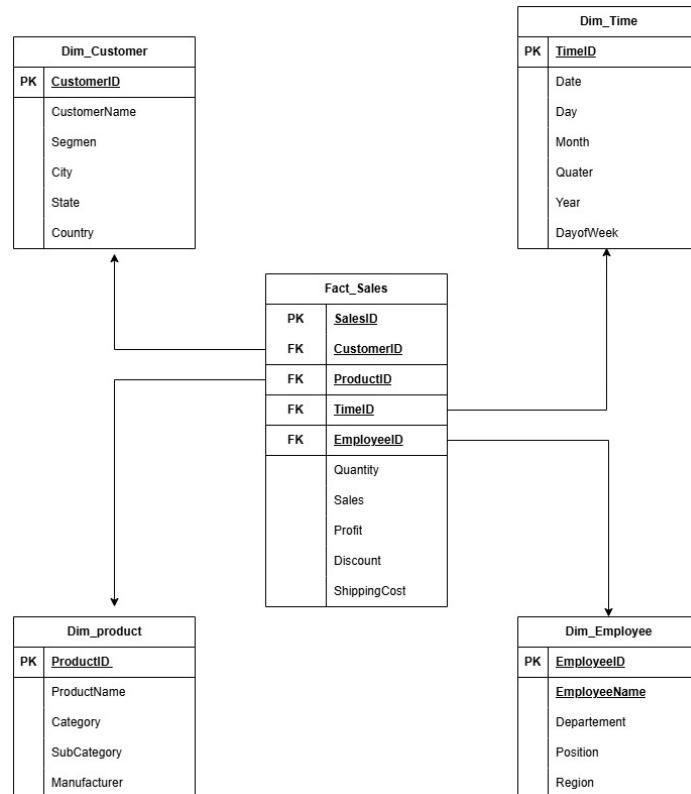
Data tersebut menggambarkan aktivitas penjualan harian selama beberapa tahun, lengkap dengan detail pelanggan serta kategori produk yang dibeli.

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.

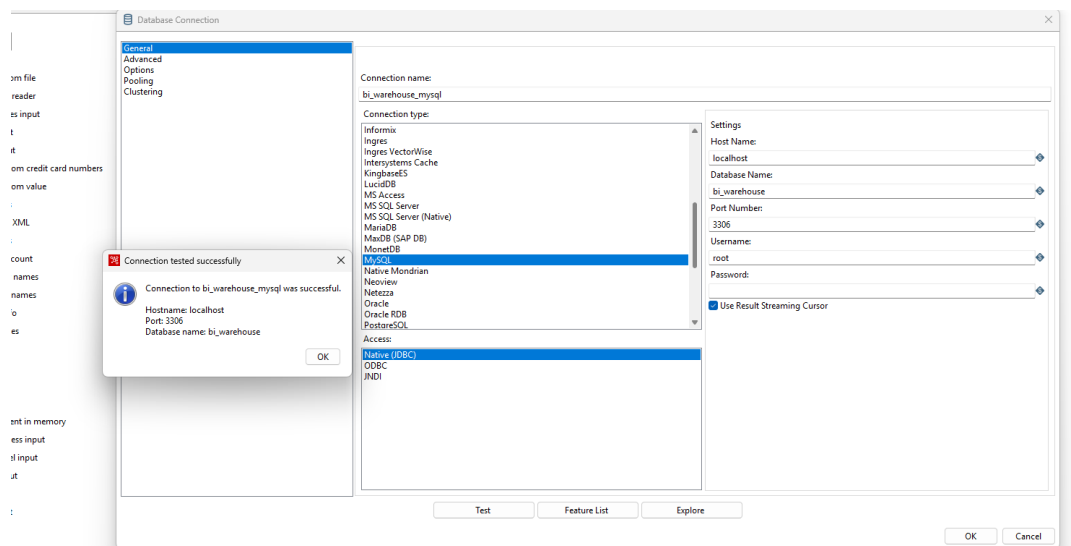
### 1.4.1 Setup Awal: Membuat Database Connection di Pentaho

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.

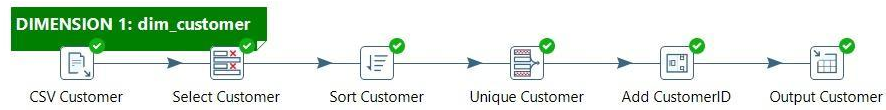
Berikut langkah-langkah pembuatan koneksi database:

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 Transformation 1: load Dim\_Customer



### 1. Komponen CSV Input

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Customer Name	String		100					both
2	Segment	String		50					both
3	City	String		100					both
4	State	String		50					both
5	Postal Code	String		20					both
6	Country	String		50					both

#### 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

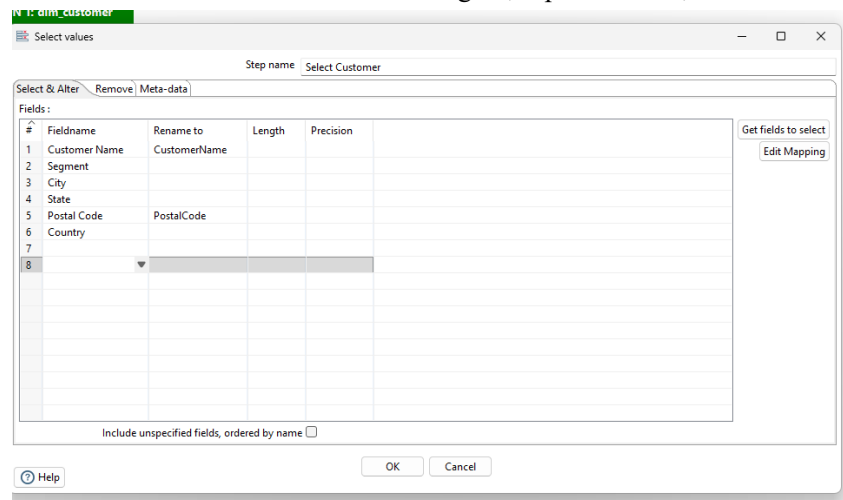
## 2. Select Values - Pilih Customer Fields

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

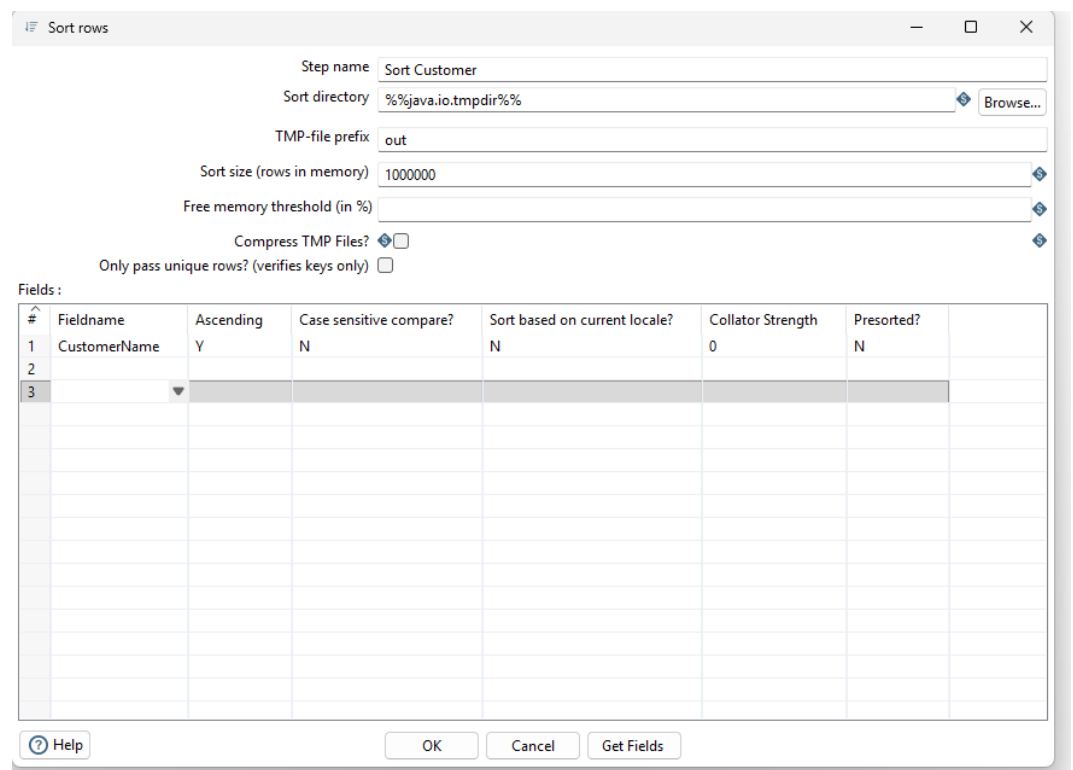


### 3. Sort Rows - Sort by Customer ID

Cara:

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

Konfigurasi:



- Fieldname: customer\_id
- Sort direction: Ascending

- c. Case sensitive: No
- d. Pre-sort rows: NO

#### 4. Unique Rows - Keep Unique Customers

Komponen ini digunakan 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

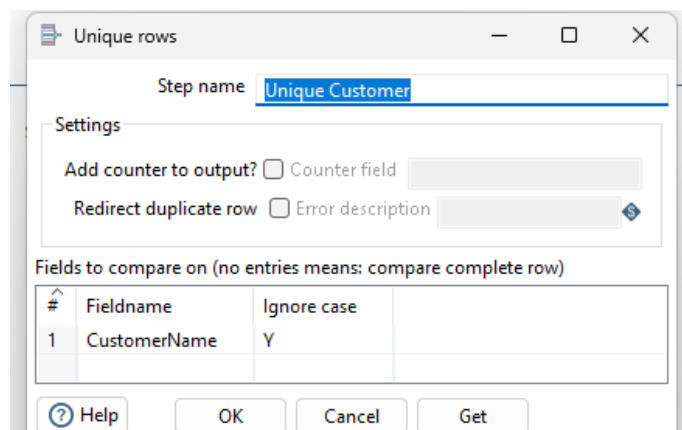
##### Konfigurasi:

Redirect duplicate row: NO (unchecked)

Error on duplicate: NO (unchecked)

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

Table output

Step name: Output Customer

Connection: bi\_warehouse\_mysql

Target schema:

Target table: dim\_customer

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

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

Table output

Step name: Output Customer

Connection: bi\_warehouse\_mysql

Target schema:

Target table: dim\_customer

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options: Database fields

Fields to insert:

#	Table field	Stream field
1	CustomerID	CustomerID
2	CustomerN...	CustomerNa...
3	Segment	Segment
4	City	City
5	State	State
6	PostalCode	PostalCode
7	Country	Country

Buttons: Get fields, Enter field mapping, Help, OK, Cancel, SQL

## 6. Preview & Execute

Setelah semua selesai lalu run dim\_customer, lalu preview datanya, seperti berikut

**Execution Results**

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	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 Kitz	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

17	Output Time	0	1020	1020	0	1020	0	0	0	Finished	0.6s	2,020
18	Output Customer	0	793	793	0	793	0	0	0	Finished	0.6s	1,365

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

`SELECT COUNT(*) FROM Dim_Customer;`

Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh

Extra options

COUNT(\*)

793

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

Showing rows 0 - 4 (5 total, Query took 0.0009 seconds)

`SELECT * FROM Dim_Customer LIMIT 5;`

Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh

Extra options

	CustomerID	CustomerName	Segment	City	State	PostalCode	Country	CreatedAt	UpdatedAt
<input type="checkbox"/> Edit Copy Delete	1	Aaron Bergman	Consumer	Seattle	Washington	98103	United States	2025-12-07 21:51:29	2025-12-07 21:51:29
<input type="checkbox"/> Edit Copy Delete	2	Aaron Hawkins	Corporate	Philadelphia	Pennsylvania	19134	United States	2025-12-07 21:51:29	2025-12-07 21:51:29
<input type="checkbox"/> 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
<input type="checkbox"/> 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
<input type="checkbox"/> 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

Check all | With selected: Edit | Copy | Delete | Export

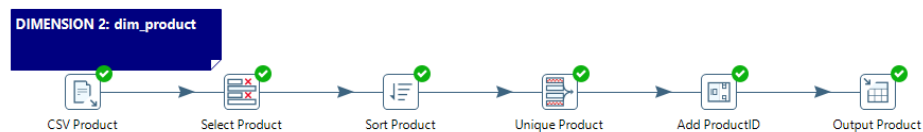
Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

### 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:

The screenshot shows the 'CSV file input' configuration window. The 'Step name' is 'CSV Product'. The 'Filename' is 'E:\smt5\BI\tubes\data.csv'. The 'Delimiter' is ',' and the 'Enclosure' is '"'. The 'NIO buffer size' is '50000'. The 'Lazy conversion?' checkbox is unchecked. The 'Header row present?' checkbox is checked. The 'Add filename to result' checkbox is unchecked. The 'The row number field name (optional)' field is empty. The 'Running in parallel?' checkbox is unchecked. The 'New line possible in fields?' checkbox is checked. The 'Format' is 'mixed' and the 'File encoding' is 'UTF-8'. The table below shows the field configurations:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Product Name	String		200					both
2	Category	String		50					both
3	Sub-Category	String		50					both
4	Sales	Number	###	15	2				both

Buttons at the bottom include 'Help', 'OK', 'Get Fields', 'Preview', and 'Cancel'.

Lakukan preview jika perlu, seperti berikut, lalu close

Examine preview data

Rows of step: CSV Product (1000 rows)

#	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	Stores 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

Close

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

Select values

Step name: **Select Product**

Select & Alter Remove Meta-data

Fields:

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

Get fields to select Edit Mapping

Include unspecified fields, ordered by name ☐

Help OK Cancel

Sort Customer : 232ms

## 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

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

[illegible]

**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.

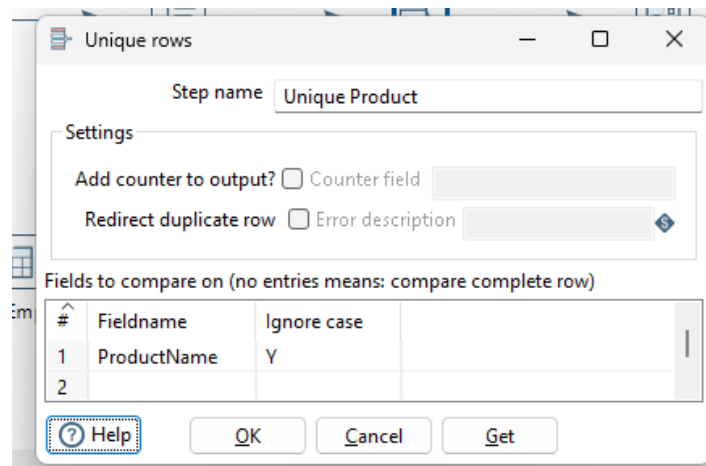
**Step name:** Unique Product

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

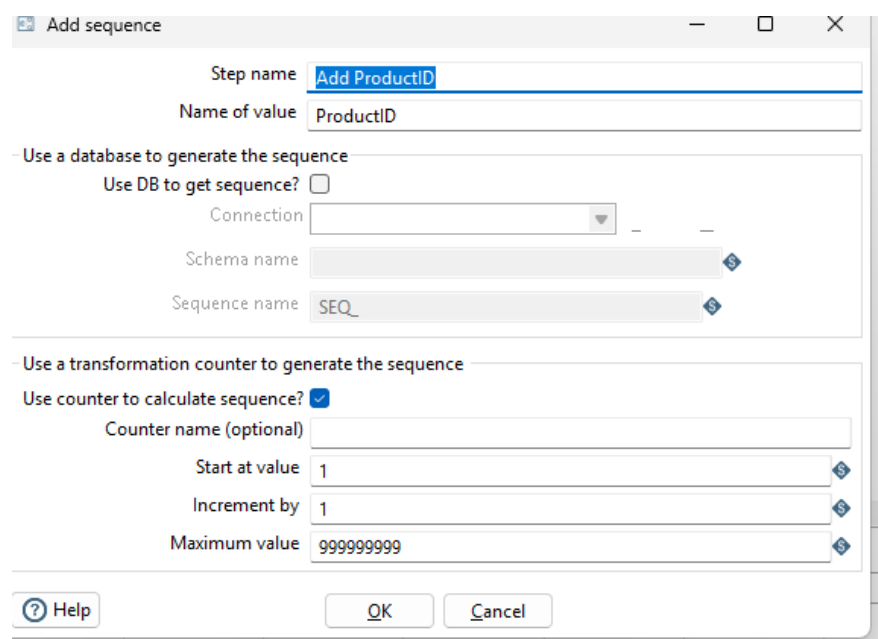


**Redirect duplicate row**, juga tidak perlu dicentang jika dicentang, baris duplikat akan diarahkan ke error handling stream, bukan dihapus.

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



#### 5. Add ProductID



#### 6. Table Output

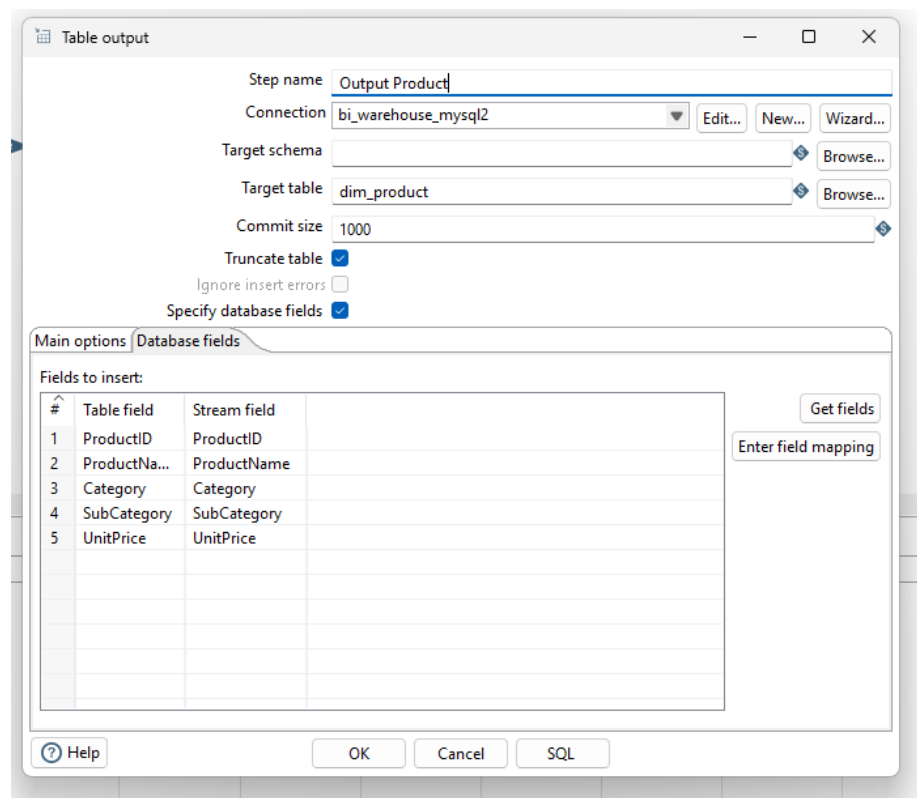
Step **Table Output** digunakan untuk **menyimpan hasil transformasi ETL ke dalam tabel di database Data Warehouse**.

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.

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Your SQL query has been executed successfully.

SELECT COUNT(\*) FROM Dim\_Product;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Extra options

COUNT(\*)

1849

Query results operations

Print Copy to clipboard Export Display chart Create view

Showing rows 0 - 9 (10 total, Query took 0.0017 seconds.)

SELECT \* FROM Dim\_Product LIMIT 10;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Extra options

	ProductID	ProductName	Category	SubCategory	UnitPrice	CreatedAt	UpdatedAt
<input type="checkbox"/> Edit Copy Delete	1	"While you Were Out" Message Book, One Form per Pa...	Office Supplies	Paper	7.42	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	2	#10 Gummed Flap White Envelopes, 100/Box	Office Supplies	Envelopes	8.26	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	3	#10 Self-Seal White Envelopes	Office Supplies	Envelopes	22.18	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	4	#10 White Business Envelopes, 4 1/8 x 9 1/2	Office Supplies	Envelopes	37.61	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	5	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	27.97	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	6	#10- 4 1/8" x 9 1/2" Security-Tint Envelopes	Office Supplies	Envelopes	24.45	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	7	#10-4 1/8" x 9 1/2" Premium Diagonal Seam Envelope...	Office Supplies	Envelopes	113.33	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	8	#6 3/4 Gummed Flap White Envelopes	Office Supplies	Envelopes	7.92	2025-12-12 10:56:45	2025-12-12 10:56:45
<input type="checkbox"/> Edit Copy Delete	9	1.7 Cubic Foot Compact "Cube" Office Refrigerators	Office Supplies	Appliances	208.16	2025-12-12 10:56:45	2025-12-12 10:56:45

route=/sql?db=supertore\_sales&table=Dim\_Product&pos=0

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

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Null if	Set empty string?
1	EmployeeID	Integer		9	0					N
2	EmployeeName	String		100						N
3	Department	String		50						N
4	Position	String		100						N
5	Region	String		50						N

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.

- 
- The screenshot shows a window titled "Data grid" with a subtitle "Step name Employee Data". The window contains a table with 6 columns: #, EmployeeID, EmployeeName, Department, Position, and Region. The table has 5 rows of data. Below the table are buttons for "Help", "OK", "Preview", and "Cancel". At the bottom of the window, there is a status bar with various numerical values.
- | # | EmployeeID | EmployeeName    | Department | Position                    | Region  |
|---|------------|-----------------|------------|-----------------------------|---------|
| 1 | 1          | John Anderson   | Sales      | Sales Representative        | East    |
| 2 | 2          | Sarah Mitchell  | Sales      | Sales Representative        | West    |
| 3 | 3          | Michael Chen    | Sales      | Senior Sales Representative | Central |
| 4 | 4          | Emily Rodriguez | Sales      | Sales Representative        | South   |
| 5 | 5          | David Thompson  | Sales      | Sales Manager               | North   |

[illegible]

14	Output product	0	1049	1049	0	1049	0	0	0	Finisnea	0.0s	2,291	-
15	Output Employee	0	5	5	0	5	0	0	0	Finished	0.0s	111	-

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

`SELECT * FROM Dim_Employee;`

EmployeeID	EmployeeName	Department	Position	Region	HireDate	CreatedAt	UpdatedAt
1	John Anderson	Sales	Sales Representative	East	NULL	2025-12-12 10:56:45	2025-12-12 10:56:45
2	Sarah Mitchell	Sales	Sales Representative	West	NULL	2025-12-12 10:56:45	2025-12-12 10:56:45
3	Michael Chen	Sales	Senior Sales Representative	Central	NULL	2025-12-12 10:56:45	2025-12-12 10:56:45
4	Emily Rodriguez	Sales	Sales Representative	South	NULL	2025-12-12 10:56:45	2025-12-12 10:56:45
5	David Thompson	Sales	Sales Manager	North	NULL	2025-12-12 10:56:45	2025-12-12 10:56:45

### 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 :**

Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1 base_date	Date	yyyy-MM-dd						2014-01-01	N

- a. Step name: Generate Dates
- b. Limit: 1826, yaitu jumlah baris yang akan di-generate nilai ini setara dengan  $\pm 5$  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.

The screenshot shows the 'Add sequence' dialog box with the following configuration:

- Step name:** Add TimeID
- Name of value:** row\_num
- Use a database to generate the sequence:**
  - Use DB to get sequence? ☐
  - Connection: [Dropdown]
  - Schema name: [Field]
  - Sequence name: SEQ\_
- Use a transformation counter to generate the sequence:**
  - Use counter to calculate sequence? ☒
  - Counter name (optional): [Field]
  - Start at value: 0
  - Increment by: 1
  - Maximum value: 99999999

Buttons: OK, Cancel, Help

### 3. Calc Date Parts

Step name: **Calc Date Parts**

Java script functions:

- Transform Scripts
- Transform Constants
- Transform Functions
- Input fields
  - base\_date
  - row\_num
- Output fields
  - Please use the 'Reg'

Script 1 X

```
// Calculate the actual date by adding row_num
var actualDate = new Date(base_date.getTime()
actualDate.setDate(actualDate.getDate() + row_r

// Extract date parts
var year = actualDate.getFullYear();
var month = actualDate.getMonth() + 1;
var day = actualDate.getDate();

// Calculate TimeID as YYYYMMDD integer
var TimeID = year * 10000 + month * 100 + day;

// Get day of week (0=Sunday, 6=Saturday)
var dayOfWeek = actualDate.getDay();
var dayNames = ["Minggu", "Senin", "Selasa", ""];
var DayOfWeek = dayNames[dayOfWeek];

// Get month name
var monthNames = ["Januari", "Februari", "Mare
var MonthName = monthNames[month - 1];

// Calculate quarter
var Quarter = Math.ceil(month / 3);

// Is weekend?
var IsWeekend = (dayOfWeek === 0 || dayOfWeek

// Format the date for output
var Date_output = actualDate;
var Day = day;
var Month = month;
var Year = year;
```

Linens: 0  
Compatibility mode? ☐ Optimization level: 9

#	Fieldname	Rename to	Type	Length	Precision	Replace va
1	TimeID		Integer	9	0	N
2	Date_output	Date	Date			N
3	Day		Integer	9	0	N
4	Month		Integer	9	0	N
5	MonthName		String	20		N
6	Quarter		Integer	9	0	N
7	Year		Integer	9	0	N
8	DayOfWeek		String	20		N
9	IsWeekend		Integer	9	0	N

Help OK Cancel Get variables Test

### 4. Tabel Output

Table output

Step name: **Output Time**

Connection: **bi\_warehouse\_mysql2** Edit... New... Wizard...

Target schema: Browse...

Target table: **dim\_time** Browse...

Commit size: 1000

Truncate table: ☒

Ignore insert errors: ☐

Specify database fields: ☐

Main options: Database fields

Fields to insert:

#	Table field	Stream field
1	TimeID	TimeID
2	Date	Date
3	Day	Day
4	Month	Month
5	MonthName	MonthName
6	Quarter	Quarter
7	Year	Year
8	DayOfWeek	DayOfWeek
9	IsWeekend	IsWeekend

Get fields

Enter field mapping

Help OK Cancel SQL



## 1.4.6 Transformasi Fakta

### 1. CSV file Input

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Order ID	String		50					both
2	Order Date	Date	dd/MM/yyyy						none
3	Customer Name	String		100					both
4	Region	String		50					both
5	Product Name	String		200					both
6	Sales	Number	#,##	15	2				none

### 2. Select Values

#	Fieldname	Rename to	Length	Precision
1	Order ID	order_id		
2	Order Date	order_date		
3	Customer Name	customer_name		
4	Region	region		
5	Product Name	product_name		
6	Sales	sales		

### 3. Generate Values

Modified JavaScript value

Step name: **Generate Values**

Java script functions:

- Transform Scripts
- Transform Constants
- Transform Functions
- Input fields
  - order\_id
  - order\_date
  - customer\_name
  - region
  - product\_name
  - sales
- Output fields
  - Please use the 'Reg

Java script:

```
// Generate dummy values for missing columns in source data
var quantity = Math.floor(Math.random() * 10) + 1; // Random 1-10
var discount = Math.floor(Math.random() * 4) * 0.1; // 0, 0.1, 0.2, or 0.3
var profit = sales * 0.15; // 15% profit margin
var shipping_cost = sales * 0.05 + 5; // 5% of sales + $5 base
```

Linens: 0

Compatibility mode? ☐ Optimization level: **g**

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	quantity		Integer	9	0	N
2	discount		Number	5	2	N
3	profit		Number	15	2	N
4	shipping_cost		Number	15	2	N

Buttons: Help, OK, Cancel, Get variables, Test script

### 4. Lookup Customer

Database lookup

Step name: **Lookup Customer**

Connection: **bi\_warehouse\_mysql** Edit... New... Wizard...

Lookup schema: Browse...

Lookup table: **dim\_customer** Browse...

Enable cache? ☒

Cache size in rows (0=cache): **0**

Load all data from table ☒

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	CustomerName	=	customer_name	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	CustomerID	customer_id		Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

## 5. Lookup Product

The 'Database lookup' window is configured with the following details:

- Step name:** Lookup Product
- Connection:** bi\_warehouse\_mysql
- Lookup schema:** (empty)
- Lookup table:** dim\_product
- Enable cache?** ☒
- Cache size in rows (0=cache):** 0
- Load all data from table:** ☒
- The key(s) to look up the value(s):**

#	Table field	Comparator	Field1	Field2
1	ProductName	=	product_name	
- Values to return from the lookup table:**

#	Field	New name	Default	Type
1	ProductID	product_id		Integer
- Do not pass the row if the lookup fails:** ☐
- Fail on multiple results?:** ☐
- Order by:** (empty)

Buttons at the bottom: Help, OK, Cancel, Get Fields, Get lookup fields.

## 6. Lookup Employee

The 'Transformation properties' window shows the following configuration:

- Transformation name:** TR\_Load\_Fact\_Sales\_v2
- Transformation filename:** E:\smt5\BI\tubes\TR\_Load\_Fact\_Sales.ktr
- Description:** Load Fact Sales with all INTEGER IDs - Run AFTER TR\_Load\_Dimensions
- Extended description:** (empty text area)
- Status:** (dropdown menu)
- Version:** (text field)
- Directory:** /
- Created by:** -
- Created at:** Sat Dec 07 21:44:00 ICT 2024
- Last modified by:** -
- Last modified at:** Sat Dec 07 21:44:00 ICT 2024

Buttons at the bottom: OK, SQL, Cancel.

## 7. Lookup Time

The 'Database lookup' window is configured with the following settings:

- Step name: **Lookup Time**
- Connection: **bi\_warehouse\_mysql**
- Lookup schema: (empty)
- Lookup table: **dim\_time**
- Enable cache?: ☒
- Cache size in rows (0=cache): **0**
- Load all data from table: ☒

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	Date	=	order_date	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	TimeID	time_id		Integer

Do not pass the row if the lookup fails: ☐  
Fail on multiple results?: ☐  
Order by: (empty)

Buttons: ? Help, OK, Cancel, Get Fields, Get lookup fields

## 8. Filter Valid

The 'Filter rows' window is configured with the following settings:

- Step name: **Filter Valid**
- Send 'true' data to step: **Table Output**
- Send 'false' data to step: (empty)

The condition:

☐ +

`customer_id IS NOT NULL`

AND

`product_id IS NOT NULL`

AND

`employee_id IS NOT NULL`

AND

`time_id IS NOT NULL`

Buttons: ? Help, OK, Cancel

## 9. Table Output

Table output

Step name: **Table Output**

Connection: **bi\_warehouse\_mysql** [Edit... New... Wizard...]

Target schema: [Browse...]

Target table: **fact\_sales** [Browse...]

Commit size: **1000**

Truncate table: ☒

Ignore insert errors: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	OrderID	order_id
2	CustomerID	customer_id
3	ProductID	product_id
4	EmployeeID	employee_id
5	TimeID	time_id
6	Quantity	quantity
7	Sales	sales
8	Profit	profit
9	Discount	discount
10	ShippingC...	shipping_cost

[Get fields] [Enter field mapping]

[?] Help [OK] [Cancel] [SQL]

## 10. Hasil saat semua dihubungkan dan dirun

FACT SALES

CSV Input → Select Values → Generate Values → Lookup Customer → Lookup Product → Lookup Employee → Lookup Time → Filter Valid → Table Output

Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	order_id	order_date	customer_name	region	product_name	sales
1	CA-2017-152156	08/11/2017	Claire Gule	South	Bush Somerset Collection Bookcase	281.96
2	CA-2017-152156	08/11/2017	Claire Gule	South	Hem Deluxe Fabric Upholstered Stacking Chairs, Rounded Back	791.94
3	CA-2017-138688	12/06/2017	Darvin Van Huff	West	Self-Adhesive Address Labels for Typewriters by Universal	14.62
4	US-2016-108966	11/10/2016	Sean O'Donnell	South	Bretford CRA500 Series Slim Rectangular Table	957.58
5	US-2016-108966	11/10/2016	Sean O'Donnell	South	Eldon Fold N Roll Cart System	22.37
6	CA-2015-115812	09/06/2015	Brosina Hoffman	West	Eldon Expressions Wood and Plastic Desk Accessories, Cherry Wood	48.86
7	CA-2015-115812	09/06/2015	Brosina Hoffman	West	Newell 322	7.28
8	CA-2015-115812	09/06/2015	Brosina Hoffman	West	Mitel 5320 IP Phone VoIP phone	907.15
9	CA-2015-115812	09/06/2015	Brosina Hoffman	West	DLL Angle-View Binders with Locking Rings by Samall	18.5
10	CA-2015-115812	09/06/2015	Brosina Hoffman	West	Ballou PSC200VTEL 6 Outlet Surge	114.9
11	CA-2015-115812	09/06/2015	Brosina Hoffman	West	Chromcraft Rectangular Conference Tables	1706.18
12	CA-2015-115812	09/06/2015	Brosina Hoffman	West	Konfel 250 Conference phone - Charcoal black	911.42
13	CA-2018-144412	15/04/2018	Andrew Allen	South	Xerox 1607	15.55
14	CA-2017-161389	09/12/2017	Irene Mendez	West	Fellowes PE200 Plastic Comb Binding Machine	402.98
15	US-2016-118983	22/11/2016	Harold Paulan	Central	Holmes Replacement Filter for HEPA Air Cleaner, Very Large Room, HEPA Filter	68.81
16	US-2016-118983	22/11/2016	Harold Paulan	Central	Score DuraTech Recycled Plastic Fronted Binders	2.54
17	CA-2015-105883	11/11/2015	Pete Kite	Central	Star-D-Store Shelving, Vertical 5-Shelf: 72" H x 36" W x 18 1/2" D	665.88
18	CA-2015-167164	13/05/2015	Alexandro Grove	West	Fellowes Super Star Drawer	55.5
19	CA-2015-143336	27/08/2015	Zuschuss Donatelli	West	Newell 341	8.56
20	CA-2015-143336	27/08/2015	Zuschuss Donatelli	West	Cisco SPA 501G IP Phone	213.48
21	CA-2015-143336	27/08/2015	Zuschuss Donatelli	West	Wilson Jones Hanging View Binder, White, 1"	22.72
22	CA-2017-137330	09/12/2017	Ken Black	Central	Newell 318	19.46
23	CA-2017-137330	09/12/2017	Ken Black	Central	Acco Six-Outlet Power Strip, 4' Cord Length	60.34

## 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

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import re
5 from pathlib import Path
6 import ast
7
8 # 1. Siapkan path file .sql yang berisi dump data warehouse (MySQL dump dari phpMyAdmin)
9 sql_file_path = Path('/Users/ahmd.rmdhnn/Projects/BI/superstore_sales.sql')
10
11 if not sql_file_path.exists():
12     raise FileNotFoundError(f'File SQL tidak ditemukan: {sql_file_path}')
13
14 # 2. Fungsi untuk parsing MySQL dump
15 def parse_mysql_dump_simple(sql_file_path):
16     """Parse MySQL dump file dengan pendekatan sederhana"""
17     with open(sql_file_path, 'r', encoding='utf-8') as f:
18         content = f.read()
19
20     tables = {}
21
22     # Cari semua INSERT statements beserta kolomnya
23     # Pattern: INSERT INTO `table` (`col1`, `col2`, ...) VALUES
24     header_pattern = r"INSERT INTO `(\w+)` `s*\((([^\)]+)\)`\s*VALUES?"
25
26     # Temukan semua header INSERT
27     for header_match in re.finditer(header_pattern, content):
28         table_name = header_match.group(1)
29         columns = [col.strip().strip('`') for col in header_match.group(2).split(',')
30
31         if table_name not in tables:
32             tables[table_name] = {'columns': columns, 'rows': []}
33
34     # Sekarang cari semua tuple data (xxx, xxx, xxx)
35     # Kita parse setelah VALUES sampai semicolon
36
37     # Split content by INSERT INTO untuk proses per statement
38     insert_blocks = re.split(r'INSERT INTO `(\w+)`', content)
```



```
39
40 i = 1
41 while i < len(insert_blocks):
42     table_name = insert_blocks[i]
43     if i + 1 < len(insert_blocks):
44         block = insert_blocks[i + 1]
45
46     # Cari kolom
47     col_match = re.search(r'\(((^)+)\)\s*VALUES?', block)
48     if col_match:
49         columns = [c.strip().strip(' ') for c in col_match.group(1).split(',')]
50
51     # Cari semua rows - semuanya setelah VALUES sampai ;
52     values_start = block.find('VALUES')
53     if values_start == -1:
54         values_start = block.find('VALUE')
55
56     if values_start != -1:
57         # Ambil semua setelah VALUES/VALUE
58         values_part = block[values_start:]
59         end_pos = values_part.find(';')
60         if end_pos != -1:
61             values_part = values_part[:end_pos]
62
63     # Parse setiap row tuple
64     rows = extract_rows(values_part)
65
66     if table_name not in tables:
67         tables[table_name] = {'columns': columns, 'rows': []}
68
69     tables[table_name]['columns'] = columns
70     tables[table_name]['rows'].extend(rows)
71     i += 2
72
73 # Konversi ke DataFrames
74 result = {}
75 for name, data in tables.items():
76     if data['rows']:
77         # Filter rows yang panjangnya sesuai dengan kolom
78         valid_rows = [r for r in data['rows'] if len(r) == len(data['columns'])]
79         if valid_rows:
80             result[name] = pd.DataFrame(valid_rows, columns=data['columns'])
81
82 return result
```

```

84 def extract_rows(values_str):
85     """Ekstrak semua tuple rows dari string VALUES"""
86     rows = []
87
88     # Cari semua pattern (...)
89     i = 0
90     while i < len(values_str):
91         if values_str[i] == '(':
92             # Mulai parsing satu tuple
93             row_values = []
94             current_val = ""
95             in_string = False
96             j = i + 1
97
98             while j < len(values_str):
99                 char = values_str[j]
100
101                 if char == '\\' and j + 1 < len(values_str):
102                     # Escape character - tambahkan karakter berikutnya
103                     current_val += values_str[j + 1]
104                     j += 2
105                     continue
106
107                 if char == '"' and not in_string:
108                     in_string = True
109                 elif char == '"' and in_string:
110                     # Cek apakah ini escaped quote ''
111                     if j + 1 < len(values_str) and values_str[j + 1] == '"':
112                         current_val += '"'
113                         j += 2
114                         continue
115                     in_string = False
116                 elif char == ',' and not in_string:
117                     row_values.append(clean_value(current_val.strip()))
118                     current_val = ""
119                     j += 1
120                     continue
121                 elif char == ')' and not in_string:
122                     # Akhir tuple
123                     row_values.append(clean_value(current_val.strip()))
124                     rows.append(row_values)
125                     i = j
126                     break
127
128                 current_val += char
129                 j += 1
130             i += 1
131
132     return rows

```



```

134 def clean_value(val):
135     """Bersihkan dan konversi nilai dari SQL"""
136     if val.upper() == 'NULL':
137         return None
138     if val.startswith('"') and val.endswith('"'):
139         return val[1:-1]
140     try:
141         if '.' in val:
142             return float(val)
143         return int(val)
144     except:
145         return val
146
147 # 3. Parse file SQL
148 print("Memproses file SQL...")
149 tables = parse_mysql_dump_simple(sql_file_path)
150
151 print("\nTabel yang ditemukan:")
152 for name, tbl in tables.items():
153     print(f" - {name}: {len(tbl)} baris, kolom: {list(tbl.columns)}")
154
155 # 4. Gabungkan tabel-tabel
156 fact_sales = tables.get('fact_sales', pd.DataFrame())
157 dim_time = tables.get('dim_time', pd.DataFrame())
158 dim_product = tables.get('dim_product', pd.DataFrame())
159 dim_customer = tables.get('dim_customer', pd.DataFrame())
160 dim_employee = tables.get('dim_employee', pd.DataFrame())
161
162 # Merge semua tabel dengan pengecekan
163 df = fact_sales.copy()
164
165 if not df.empty:
166     if not dim_time.empty and 'TimeID' in df.columns and 'TimeID' in dim_time.columns:
167         df = df.merge(dim_time, on='TimeID', how='left', suffixes=('', '_time'))
168     if not dim_product.empty and 'ProductID' in df.columns and 'ProductID' in dim_product.columns:
169         df = df.merge(dim_product, on='ProductID', how='left', suffixes=('', '_product'))
170     if not dim_customer.empty and 'CustomerID' in df.columns and 'CustomerID' in dim_customer.columns:
171         df = df.merge(dim_customer, on='CustomerID', how='left', suffixes=('', '_customer'))
172     if not dim_employee.empty and 'EmployeeID' in df.columns and 'EmployeeID' in dim_employee.columns:
173         df = df.merge(dim_employee, on='EmployeeID', how='left', suffixes=('', '_employee'))
174
175 # Konversi kolom Date ke datetime
176 if 'Date' in df.columns:
177     df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
178     df['YearMonth'] = df['Date'].dt.to_period('M')
179
180 print("\nData berhasil dimuat. Jumlah baris:", len(df))
181 print("\nKolom tersedia:", list(df.columns))
182 print("\nPreview data:")
183 print(df.head())

```

## 2. Membuat visualisasi untuk masing-masing KPI

### a. Tren Bulanan

```
BI - visualisasi.ipynb

1 # Agregasi data berdasarkan Bulan
2 monthly_sales = df.groupby('YearMonth')['Sales'].sum().reset_index()
3 monthly_sales['YearMonth'] = monthly_sales['YearMonth'].astype(str)
4
5 plt.figure(figsize=(15, 6))
6 sns.lineplot(data=monthly_sales, x='YearMonth', y='Sales', marker='o', color='#1f77b4')
7
8 plt.title('Monthly Revenue Trend', fontsize=16)
9 plt.xlabel('YearMonth', fontsize=12)
10 plt.ylabel('Sales', fontsize=12)
11 plt.xticks(rotation=45, fontsize=10)
12 plt.grid(True, linestyle='--', alpha=0.6)
13 plt.tight_layout()
14 plt.show()
```

### b. Produk dengan pendapatan terbesar

```
BI - visualisasi.ipynb

1 # Agregasi data berdasarkan Nama Produk, urutkan descending, ambil 10 teratas
2 top_products = df.groupby('ProductName')['Sales'].sum().sort_values(ascending=False).head(10).reset_index()
3
4 plt.figure(figsize=(12, 8))
5 sns.barplot(data=top_products, y='ProductName', x='Sales', palette='Blues_r')
6
7 plt.title('Top 10 Products by Revenue Contribution', fontsize=16)
8 plt.xlabel('Total Revenue', fontsize=12)
9 plt.ylabel('Product Name', fontsize=12)
10 plt.grid(axis='x', linestyle='--', alpha=0.6)
11 plt.tight_layout()
12 plt.show()
```

### c. Total penjualan berdasarkan segmen customer

```
BI - visualisasi.ipynb

1 # Agregasi berdasarkan Segment
2 segment_sales = df.groupby('Segment')['Sales'].sum().reset_index()
3
4 plt.figure(figsize=(10, 6))
5 colors = ['#1f77b4', '#d62728', '#2ca02c']
6 sns.barplot(data=segment_sales, x='Segment', y='Sales', palette=colors)
7
8 plt.title('Total Revenue by Customer Segment', fontsize=16)
9 plt.xlabel('Segment', fontsize=12)
10 plt.ylabel('Revenue', fontsize=12)
11 plt.grid(axis='y', linestyle='--', alpha=0.6)
12 plt.tight_layout()
13 plt.show()
```

### d. Performa penjualan antar wilayah

```
BI - visualisasi.ipynb

1 # Agregasi berdasarkan Region
2 region_sales = df.groupby('Region')['Sales'].sum().reset_index()
3
4 plt.figure(figsize=(10, 6))
5 colors = ['#1f77b4', '#d69e2e', '#0f9d58', '#c0504d']
6 sns.barplot(data=region_sales, x='Region', y='Sales', palette=colors)
7
8 plt.title('Total Sales by Region', fontsize=16)
9 plt.xlabel('Region', fontsize=12)
10 plt.ylabel('Sales', fontsize=12)
11 plt.grid(axis='y', linestyle='--', alpha=0.6)
12 plt.tight_layout()
13 plt.show()
```

e. Margin keuntungan tiap kategori produk

```
BI - visualisasi.ipynb

1 # Agregasi berdasarkan Category untuk Profit
2 category_profit = df.groupby('Category')['Profit'].sum().reset_index()
3
4 plt.figure(figsize=(10, 6))
5 colors = ['#1f77b4', '#d69e2e', '#0f9d58']
6 sns.barplot(data=category_profit, x='Category', y='Profit', palette=colors)
7
8 plt.title('Total Profit by Category', fontsize=16)
9 plt.xlabel('Category', fontsize=12)
10 plt.ylabel('Profit', fontsize=12)
11 plt.grid(axis='y', linestyle='--', alpha=0.6)
12 plt.tight_layout()
13 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.

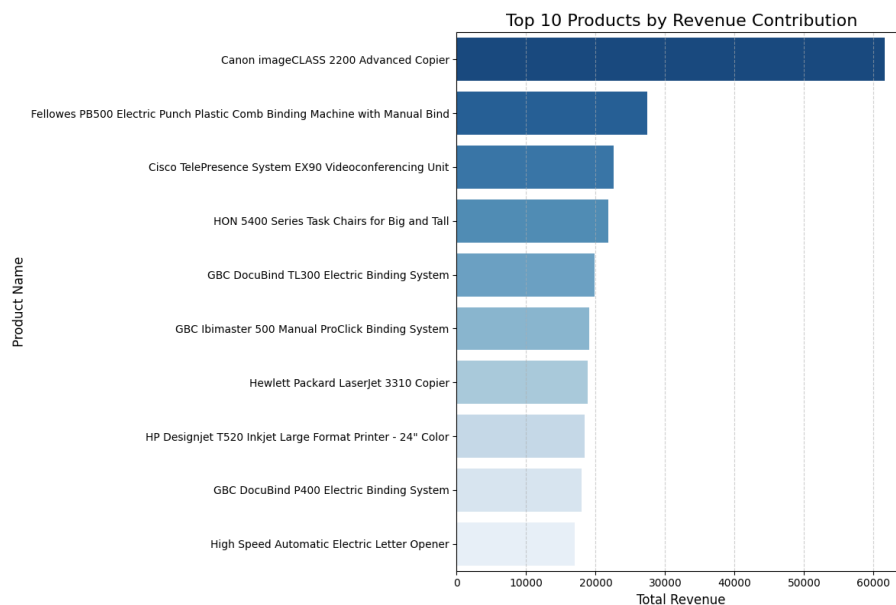


Grafik garis ini menggambarkan total pendapatan penjualan (sales) yang dikelompokkan berdasarkan bulan dan tahun (YearMonth).

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.

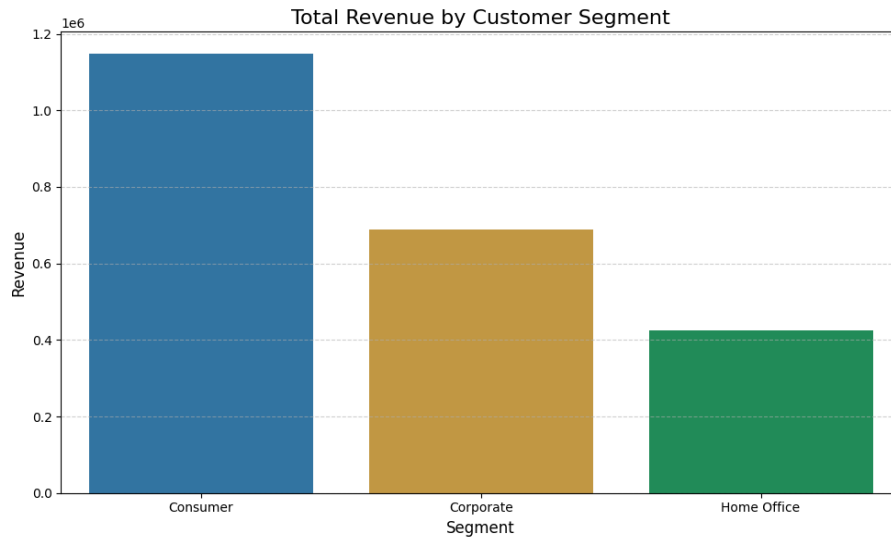
## 2. Produk dengan kontribusi pendapatan terbesar.



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.

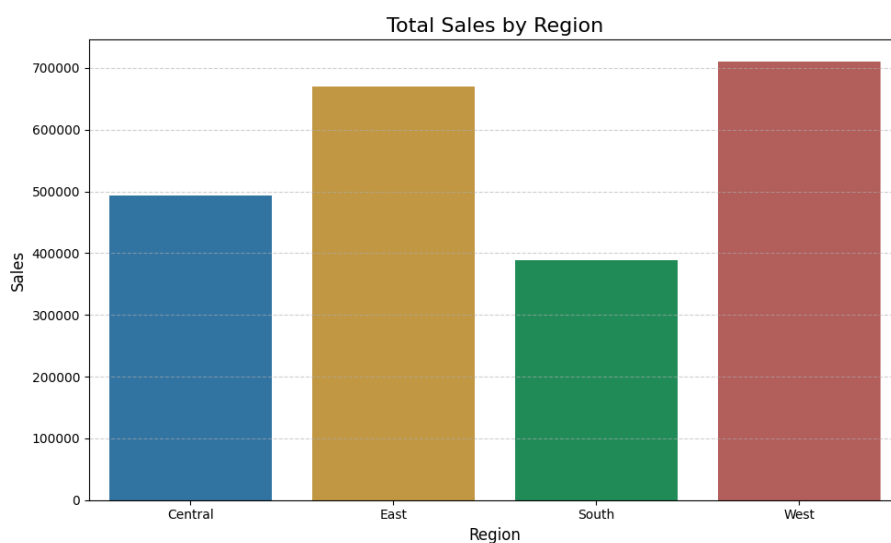
## 3. Proporsi penjualan berdasarkan segmen pelanggan.



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.

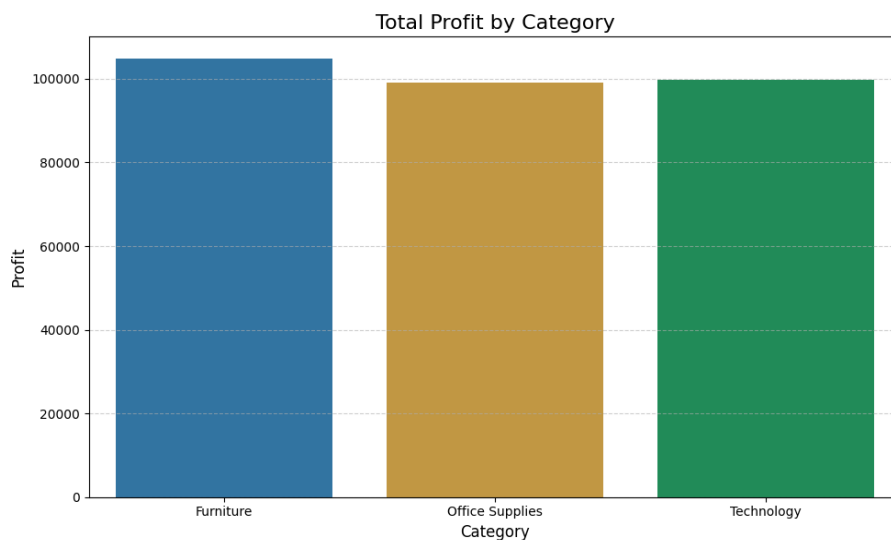
#### 4. Perbandingan performa penjualan antar wilayah.



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>