

## **TUGAS AKHIR DWO**

### **LAPORAN IMPLEMENTASI MEMBUAT DASHBOARD DAN MONDRIAN DENGAN DATA PRODUK DAN SALES DARI ADVENTUREWORKS**

**Dosen Pengampu:**

MOHAMAD IRWAN AFANDI, ST, M.Sc.



**Disusun Oleh**

Alfathur Robbani

22082010194

Muhammad Rafli Alfarisqi

22082010213

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR  
SURABAYA  
2024/2025**

## KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karuniaNya, kami dapat menyelesaikan laporan proyek akhir yang berjudul "Pengembangan Data Warehouse dan Dashboard". Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah Data Warehouse & Olap di Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur. Penyusunan laporan ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, kami ingin mengucapkan terima kasih kepada:

1. Bapak **MOHAMAD IRWAN AFANDI, ST.,M.Sc**, selaku dosen pengampu mata kuliah Data Warehouse & Olap yang telah memberikan bimbingan dan arahan dalam penyusunan laporan ini.
2. Rekan tim proyek akhir, yang telah bekerja sama dengan penuh dedikasi dan semangat untuk menyelesaikan laporan ini

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca.

Surabaya, 26-12-2024

Penyusun

## DAFTAR ISI

<b>COVER.....</b>	<b>1</b>
<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>BAB 1.....</b>	<b>5</b>
<b>PENDAHULUAN.....</b>	<b>5</b>
1.1 Latar Belakang.....	5
1.2 Tujuan.....	7
1.3 Manfaat.....	7
<b>BAB 2.....</b>	<b>8</b>
<b>METODOLOGI.....</b>	<b>8</b>
2.1 Metodologi.....	8
2.1.1 Analisis Adventureworks.....	8
2.1.2 Star Schema Model.....	8
2.1.3 Perancangan Data Warehouse.....	9
A. Mengidentifikasi Proses Bisnis (Identify the Business Process).....	9
B. Menentukan Granularitas Data (Determine the Grain).....	9
C. Mengidentifikasi Dimensi (Identify the Dimensions).....	9
D. Mengidentifikasi Fakta (Identify the Facts).....	10
E. Mendefinisikan Dimensi (Store the Predefined Dimensions).....	11
F. Mendesain Tabel Fakta (Store the Predefined Facts).....	11
G. Membangun Prototipe (Build the Physical Design).....	11
H. Mengelola dan Memuat Data (Load and Clean the Data).....	11
I. Iterasi dan Penyempurnaan (Iterate and Enhance).....	12
2.1.4 ETL.....	12
2.1.5 OLAP.....	13
2.1.6 IMPLEMENTASI DASHBOARD.....	13
2.2 Landasan Teori.....	13
<b>BAB 3.....</b>	<b>19</b>
<b>HASIL &amp; PEMBAHASAN.....</b>	<b>19</b>
3.1 Implementasi Multidimensi.....	19
3.2 Analisis kebutuhan data tabel dimensi.....	22
3.3 Tabel Fakta.....	34
3.4 Relevansi Tabel fakta dengan tabel dimensi.....	35
3.5 Proses ETL & OLAP.....	36
3.6 Implementasi Dashboard & Hasil Insight Analisis.....	40
3.6.1 Analisis Pendapatan Hasil Kerja Karyawan.....	40
3.6.2 Analisis total penjualan berdasarkan kategori.....	41
3.6.2 Analisis Data Penjualan Setiap Kategori Produk.....	44

3.6.3 Analisis Data Barang Terlaris.....	45
3.6.3 Analisis Pendapatan Setiap Toko Berdasarkan Wilayah.....	46
<b>BAB 4.....</b>	<b>48</b>
<b>KESIMPULAN &amp; SARAN.....</b>	<b>48</b>
<b>Lampiran.....</b>	<b>49</b>

# BAB 1

## PENDAHULUAN

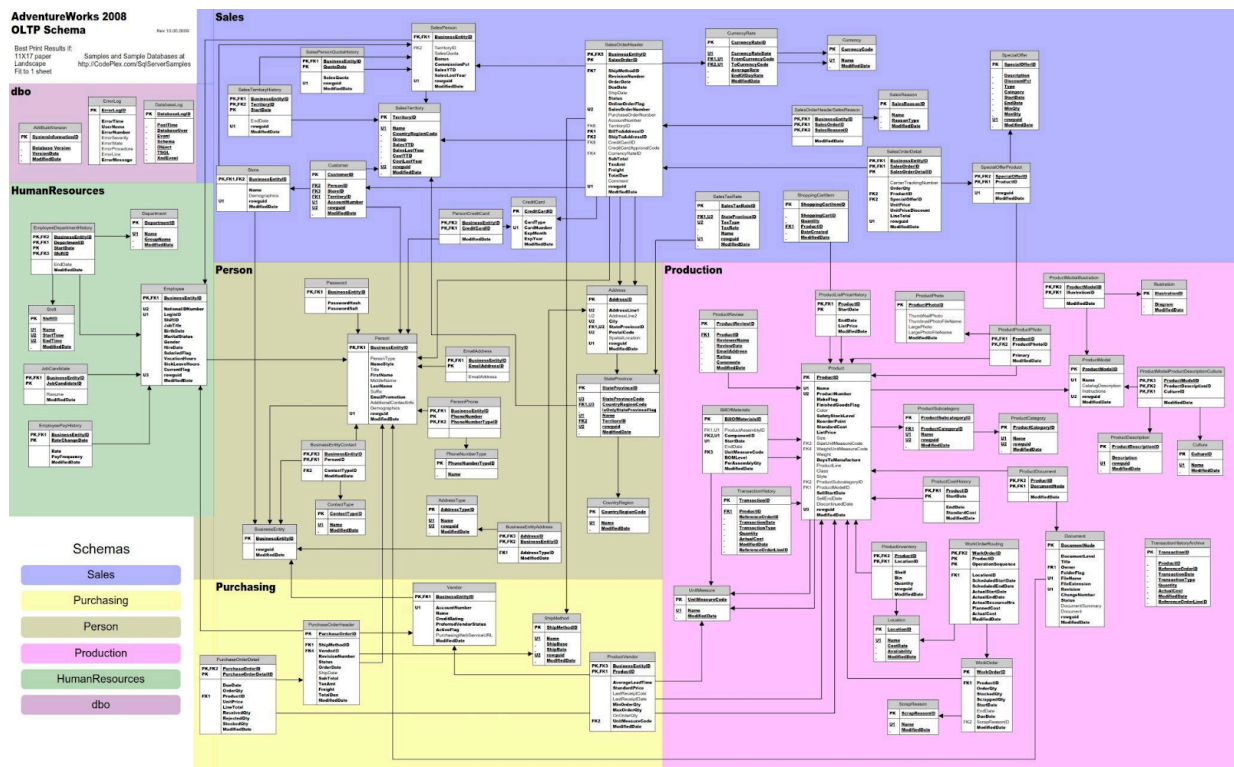
### 1.1 Latar Belakang

Dalam era digitalisasi dan perkembangan teknologi informasi, data menjadi salah satu aset penting yang dimanfaatkan oleh berbagai perusahaan untuk menganalisis dan meningkatkan kinerja bisnis. Salah satu contoh penerapan pengelolaan data ini adalah dalam industri manufaktur dan penjualan, di mana perusahaan menggunakan data untuk mengelola proses produksi, penjualan, distribusi, hingga hubungan dengan pelanggan.

Adventure Works adalah database simulasi yang dirancang oleh Microsoft untuk mendemonstrasikan fitur-fitur SQL Server. Database ini sering digunakan sebagai studi kasus dalam pengajaran dan penelitian terkait pengelolaan data bisnis. Adventure Works mencakup data perusahaan fiktif yang bergerak di bidang produksi dan penjualan produk olahraga, seperti sepeda, aksesoris sepeda, dan peralatan olahraga lainnya. Struktur database ini mencakup berbagai tabel yang mencerminkan skenario bisnis nyata, sehingga cocok digunakan sebagai sumber data untuk pengembangan proyek analisis bisnis dan pembuatan data warehouse.

Dalam proyek ini, database **Adventure Works 2019** digunakan sebagai sumber data utama untuk membangun data warehouse sederhana yang bertujuan menganalisis penjualan, kinerja karyawan, dan wilayah penjualan. Beberapa tabel utama dari database Adventure Works 2019 diintegrasikan ke dalam data warehouse yang dibangun untuk mendukung kebutuhan analisis tersebut.

Dengan menggunakan tabel-tabel utama dari database Adventure Works, data warehouse ini memungkinkan pengambilan keputusan yang lebih baik melalui analisis data berbasis fakta. Proyek ini juga bertujuan untuk mendemonstrasikan bagaimana data dari sistem operasional dapat diubah menjadi informasi yang berguna melalui proses ETL (Extract, Transform, Load) ke dalam data warehouse.



Gambar 1.1 Struktur Database AdventureWorks

Database Adventureworks memiliki banyak tabel yang mencerminkan berbagai aspek bisnis, seperti data pelanggan, produk, karyawan, wilayah, dan transaksi penjualan. Beberapa tabel utama yang digunakan sebagai sumber data dalam proyek ini adalah tabel sales\_customer, yang menyimpan informasi pelanggan perusahaan, termasuk kolom utama seperti CustomerID (ID pelanggan) dan PersonID yang menghubungkan data pelanggan dengan data personal di tabel lain. Selain itu, tabel person\_person menyimpan informasi individu terkait, seperti pelanggan dan karyawan, dengan kolom penting seperti BusinessEntityID, FirstName, MiddleName, dan LastName.

Tabel humanresources\_employee berisi data karyawan perusahaan, mencakup informasi seperti jenis kelamin (Gender), status perkawinan (MaritalStatus), dan tanggal lahir (BirthDate). Untuk data produk, tabel production\_product menyimpan informasi produk yang dijual, termasuk nama produk (Name) dan kategorinya. Kategori ini dijelaskan lebih rinci melalui tabel production\_productsubcategory dan production\_productcategory, yang masing-masing memberikan data subkategori dan kategori utama dari produk.

Wilayah penjualan perusahaan dicatat dalam tabel sales\_salesterritory, yang memiliki kolom seperti TerritoryID dan Name untuk mengidentifikasi wilayah tertentu. Sedangkan data transaksi penjualan dicatat dalam tabel sales\_salesorderdetail dan sales\_salesorderheader, yang memuat informasi penting seperti SalesOrderID, OrderDate, TotalDue, dan ProductID

## **1.2 Tujuan**

1. Mengembangkan information dari database Adventureworks 2019 yang relevan dan terintegrasi untuk keperluan analisis penjualan dan pembuatan produk.
2. Membuat dashboard sebagai alat visualisasi information untuk mendukung pengambilan keputusan yang cepat dan akurat.
3. Memberikan knowledge mendalam tentang pola penjualan produk dan pembelian oleh tim deals melalui analisis information berbasis dashboard.

## **1.3 Manfaat**

1. Kemudahan Analisis Data, Dengan data warehouse yang dirancang khusus, proses analisis data menjadi lebih cepat, akurat, dan terorganisir, sehingga mendukung kebutuhan bisnis.
2. Peningkatan Pengambilan Keputusan, Dashboard menyediakan visualisasi yang mudah dipahami, membantu manajemen dalam mengambil keputusan yang berbasis data.
3. Efisiensi Operasional, Data warehouse mengurangi redundansi dan kompleksitas pengelolaan data, sehingga mendukung efisiensi operasional perusahaan.

## BAB 2

### METODOLOGI

#### 2.1 Metodologi



Gambar 2.1 Alur Metodologi yang Digunakan Selama Penelitian

##### 2.1.1 Analisis Adventureworks

Adventure Works adalah sebuah kasus bisnis dan database sebagai alat pedagogical untuk menyediakan banyak peluang pembelajaran yang aktif kepada siswa di database yang lebih advanced, business intelligence, dan application development classes. Pada proses ini, penelitian dilakukan dengan menganalisis OLTP dan ERD yang ada pada Adventure Works. Hasil analisis yang didapat akan dijadikan informasi yang nantinya digunakan untuk proses selanjutnya.

##### 2.1.2 Star Schema Model

Star Schema model digunakan untuk menyederhanakan skema yang tersedia dengan melakukan join antar tabel dimensi atau tabel fakta. Menurut Poe sendiri, Skema bintang merupakan rancangan yang memiliki struktur sederhana dengan tabel yang relatif dan penggabungannya yang diketahui”. Dengan menggunakan model star, perancangan multidimensi dapat dilakukan lebih mudah.



### 2.1.3 Perancangan Data Warehouse

Data warehouse merupakan sistem yang mengumpulkan data, menyimpan data, dan menganalisis data dari berbagai sumber yang ada di suatu perusahaan. Penggunaan data warehouse sangat penting agar data dari berbagai cabang menjadi satu kesatuan. Metode

yang digunakan pada penelitian ini adalah Nine Step Methodology yang dikembangkan oleh Raph Kimbal. Metode ini memiliki 9 tahapan diantaranya :

#### **A. Mengidentifikasi Proses Bisnis (Identify the Business Process)**

Dalam database Adventure Works, proses bisnis utama yang menjadi fokus adalah kegiatan penjualan, manajemen pelanggan, produk, wilayah penjualan, serta dimensi waktu. Proses ini mencerminkan aktivitas bisnis inti, seperti mencatat penjualan, memantau pelanggan, dan mengelola distribusi produk.

#### **B. Menentukan Granularitas Data (Determine the Grain)**

Granularitas data ditentukan pada tingkat transaksi individual, seperti setiap penjualan produk kepada pelanggan di wilayah tertentu pada waktu tertentu. Dalam tabel *factsales*, granularitasnya mencakup kolom SalesAmount dan QuantitySold, yang merepresentasikan rincian setiap transaksi.

#### **C. Mengidentifikasi Dimensi (Identify the Dimensions)**

Dimensi-dimensi utama diambil dari tabel-tabel terkait dalam database Adventure Works:

- dimcustomer: berisi informasi pelanggan, seperti CustomerName dan CustomerType.

- dimemployee: menyimpan data karyawan yang terlibat dalam proses bisnis, seperti EmployeeName dan MaritalStatus.
- dimproduct: memuat informasi produk, seperti ProductName dan ProductCategory.
- dimsalesterritory: mencakup wilayah penjualan dengan atribut seperti TerritoryName.
- dimtime: mencakup dimensi waktu, seperti Tahun, Bulan, dan HariJawa, yang memungkinkan analisis berdasarkan waktu.

#### D. Mengidentifikasi Fakta (Identify the Facts)

Fakta utama diletakkan dalam tabel *factsales*, yang berisi pengukuran kuantitatif seperti berikut :

##### Ringkasan Relasi Berdasarkan Pertanyaan

Pertanyaan	Jawaban	Tabel Terkait	Kolom Kunci
Apa yang diproduksi?	Produk dan kategorinya	dimproduct	ProductID , ProductName , ProductCategory
Kapan diproduksi?	Waktu produksi/transaksi	dimtime	TimeID , TanggalLengkap , HariEnglish
Dimana tempatnya?	Wilayah produksi/penjualan	dimsalesterritory	TerritoryID , TerritoryName
Siapa yang produksi?	Karyawan yang berkontribusi	dimemployee	EmployeeID , EmployeeName
Bagaimana prosesnya?	Rangkaian penjualan dari fakta	factsales	SalesAmount , Foreign Key ke tabel dimensi

### **E. Mendefinisikan Dimensi (Store the Predefined Dimensions)**

Dimensi yang telah diidentifikasi dirancang lebih lanjut:

- dimcustomer menyediakan atribut terkait pelanggan.
- dimemployee menampilkan data personal dan status karyawan.
- dimproduct berfungsi untuk mengelompokkan produk ke dalam kategori tertentu.
- dimsalesterritory memetakan wilayah geografis ke dalam sistem.
- dimtime menyajikan data waktu yang kaya, termasuk hari pasar Jawa.

### **F. Mendesain Tabel Fakta (Store the Predefined Facts)**

Tabel *factsales* dirancang sebagai inti dari data warehouse, yang menghubungkan tabel dimensi melalui kolom *foreign key* seperti ProductID, CustomerID, EmployeeID, TerritoryID, dan TimeID. Tabel ini memuat metrik-metrik penting untuk analisis.

### **G. Membangun Prototipe (Build the Physical Design)**

Prototipe sistem dibuat dengan mengimplementasikan struktur tabel dalam database MySQL berdasarkan desain yang telah dirancang. Semua tabel fakta dan dimensi dihubungkan untuk memastikan integritas data.

### **H. Mengelola dan Memuat Data (Load and Clean the Data)**

Data dari database Adventure Works di-*extract* dan di-*transform* untuk dimasukkan ke dalam tabel advuas. Contohnya:

- Data pelanggan ditransformasikan dari tabel sales\_customer dan person\_person ke dalam dimcustomer.

- Data produk diambil dari tabel `production_product`, `production_productsubcategory`, dan `production_productcategory` ke dalam `dimproduct`.

## I. Iterasi dan Penyempurnaan (Iterate and Enhance)

Setelah data warehouse diimplementasikan, proses iterasi dilakukan untuk menyempurnakan desain.

### 2.1.4 ETL

ETL (Extract, Transform, Load) adalah proses utama dalam pengolahan data yang digunakan untuk memindahkan data dari berbagai sumber ke dalam data warehouse. Proses ini terdiri dari tiga tahapan utama:

1. **Extract (Ekstraksi):** Mengambil data mentah dari berbagai sumber sistem atau database. Dalam konteks proyek ini, data diekstraksi dari database Adventure Works yang berisi tabel-tabel utama seperti `sales_customer`, `person_person`, `production_product`, dan `sales_salesterritory`.
2. **Transform (Transformasi):** Membersihkan, memformat, dan memproses data agar sesuai dengan kebutuhan analisis. Transformasi ini mencakup perubahan format, penggabungan tabel, perhitungan nilai baru, atau pengelompokan data. Misalnya, dalam proyek ini:
  - a. Data dari `sales_customer` digabung dengan `person_person` untuk mendapatkan informasi lengkap pelanggan, seperti `CustomerName` dan `CustomerType`, yang dimasukkan ke tabel `dimcustomer`.
  - b. Data produk dari `production_product`, `production_productsubcategory`, dan `production_productcategory` diolah untuk mendapatkan atribut `ProductCategory` di `dimproduct`.
3. **Load (Memuat Data):** Memasukkan data yang telah diolah ke dalam data warehouse. Pada proyek ini, data yang sudah ditransformasi dimasukkan ke dalam tabel-tabel dimensi dan fakta di database `advuas`.

### 2.1.5 OLAP

**OLAP (Online Analytical Processing)** adalah teknologi yang digunakan untuk menganalisis data secara multidimensi dengan cepat dan interaktif. OLAP memungkinkan pengguna untuk mengeksplorasi data dari berbagai sudut pandang atau dimensi, seperti waktu, produk, pelanggan, dan wilayah. Teknologi ini dirancang untuk mendukung pengambilan keputusan strategis melalui analisis data yang mendalam dan intuitif. Dalam OLAP, data dapat diolah untuk menghitung total, rata-rata, minimum, maksimum, atau agregasi lainnya, serta memungkinkan eksplorasi mendalam melalui operasi seperti **drill down** untuk melihat data secara rinci, **roll up** untuk melihat data secara agregat, **slice and dice** untuk memilih subset data tertentu, dan **pivoting** untuk mengubah perspektif analisis.

### 2.1.6 IMPLEMENTASI DASHBOARD

Implementasi Dashboard, di mana data warehouse yang telah dibangun diintegrasikan. Pada tahap ini, dashboard dirancang dengan menyusun elemen-elemen visual yang sesuai dengan kebutuhan analisis yang telah dilakukan, seperti grafik, tabel, dan lainnya. Dashboard ini dirancang agar dapat menyajikan informasi secara real-time, akurat, dan mudah dipahami oleh pengguna.

## 2.2 Landasan Teori

- **Database**

Database adalah kumpulan data yang terorganisir dengan baik dan terstruktur sedemikian rupa sehingga memungkinkan pengelolaan, penyimpanan, dan pengambilan kembali data secara efisien. Database didefinisikan sebagai suatu koleksi data yang saling berkaitan dan disusun untuk memenuhi berbagai kebutuhan. Data yang disimpan dalam database umumnya memiliki keterkaitan yang membentuk suatu sistem informasi, yang dapat digunakan untuk mendukung proses pengambilan keputusan. Secara umum, database

tidak hanya menyimpan data dalam jumlah besar, tetapi juga memungkinkan manipulasi data seperti menambah, menghapus, atau memperbarui informasi dengan cepat dan akurat. Sehingga, database menjadi komponen penting dalam pengembangan sistem informasi, baik itu untuk keperluan administratif, bisnis, maupun analisis data.

- **Data Warehouse**

Data warehouse adalah koleksi data yang mempunyai sifat subject-oriented, Integrated, nonvolatile, dan time-variant untuk mendukung proses pengambilan keputusan dalam manajemen. Menurut Mannino, data warehouse adalah tempat penyimpanan data terpusat, dimana data dari basis data operasional dan sumber lainnya diintegrasikan, dibersihkan dan diarsipkan untuk mendukung pengambilan keputusan. Menurut McLeod, data warehouse adalah sebuah sistem penyimpanan data yang berkapasitas besar, dimana data dikumpulkan dengan menambahkan record baru daripada meng-update record yang sudah ada dengan informasi baru. Data jenis ini digunakan hanya untuk proses pengambilan keputusan dan bukan untuk kegiatan operasional perusahaan sehari-hari

- **Sumber Data & Tools Yang Digunakan**

Tabel data yang digunakan dalam proyek ini adalah sebagai berikut :

- **Dimcustomer Table**

**Data:** Tabel ini berisi informasi pelanggan.

**Sumber Data:**

Data diambil dari tabel sales\_customer dan person\_person dari database AdventureWork2019.

CustomerID: ID unik untuk pelanggan.

CustomerName: Gabungan nama depan, nama tengah, dan nama belakang dari tabel person\_person.

CustomerType: Tipe pelanggan diambil dari kolom PersonType pada tabel person\_person.

### ➤ **Dimemployee Table**

**Data:** Tabel ini berisi informasi karyawan.

**Sumber Data:**

Data diambil dari tabel person\_person dan humanresources\_employee dari database AdventureWork2019.

EmployeeID: ID unik untuk karyawan.

EmployeeName: Gabungan nama depan, nama tengah, dan nama belakang dari tabel person\_person.

MaritalStatus: Status pernikahan karyawan dari tabel humanresources\_employee.

Gender: Jenis kelamin karyawan dari tabel humanresources\_employee.

Age: Perhitungan usia berdasarkan tahun kelahiran dari kolom BirthDate di tabel humanresources\_employee.

### ➤ **Dimproduct Table**

**Data:** Tabel ini berisi informasi produk.

**Sumber Data:**

Data diambil dari tabel production\_product, production\_productsubcategory, dan production\_productcategory dari database AdventureWork2019.

ProductID: ID unik untuk produk.

ProductName: Nama produk dari tabel production\_product.

ProductCategory: Kategori produk dari tabel production\_productcategory.

#### ➤ **Dimsalesterritory Table**

**Data:** Tabel ini berisi informasi wilayah penjualan.

##### **Sumber Data:**

Data diambil dari tabel sales\_salesterritory dari database AdventureWork2019.

TerritoryID: ID unik untuk wilayah penjualan.

TerritoryName: Nama wilayah dari tabel sales\_salesterritory.

#### ➤ **Dimtime Table**

**Data:** Tabel ini berisi informasi waktu (tanggal, hari, bulan, tahun, dll.).

##### **Sumber Data:**

Data diisi menggunakan prosedur isiDimTime untuk menjadikan data dari rentang tanggal tertentu (dari 2023-01-01 hingga 2023-01-31) dengan detail harian, hari dalam bahasa Inggris, Indonesia, dan Jawa.

TimeID: ID unik untuk data waktu.

Tahun: Tahun dari tanggal.

Bulan: Bulan dari tanggal.



Tanggal: Tanggal dari tanggal.

TanggalLengkap: Tanggal lengkap (tipe DATE).

HariEnglish: Nama hari dalam bahasa Inggris.

HariIndo: Nama hari dalam bahasa Indonesia.

HariJawa: Hari dalam sistem penanggalan Jawa.

### ➤ **Factsales Table**

**Data:** Tabel faktual yang menyimpan data transaksi penjualan.

#### **Sumber Data:**

Data berasal dari tabel dimcustomer, dimemployee, dimproduct, dimsalesterritory, dan dimtime sebagai referensi kunci asing (foreign keys) dari tabel terkait.

SalesID: ID unik untuk transaksi penjualan.

ProductID: Mengacu pada produk dari dimproduct.

TimeID: Mengacu pada waktu dari dimtime.

EmployeeID: Mengacu pada karyawan dari dimemployee.

CustomerID: Mengacu pada pelanggan dari dimcustomer.

TerritoryID: Mengacu pada wilayah dari dimsalesterritory.

SalesAmount: Jumlah penjualan dalam bentuk DECIMAL.

Dalam proses pengerjaan kami menggunakan beberapa tools untuk mempermudah pengerjaan proses analisis data, mulai dari memasukkan data atau melakukan re-store data, pengolahan data, hingga visualisasi hasil analisis. Berikut tools yang kami gunakan dan alasan pemilihannya.

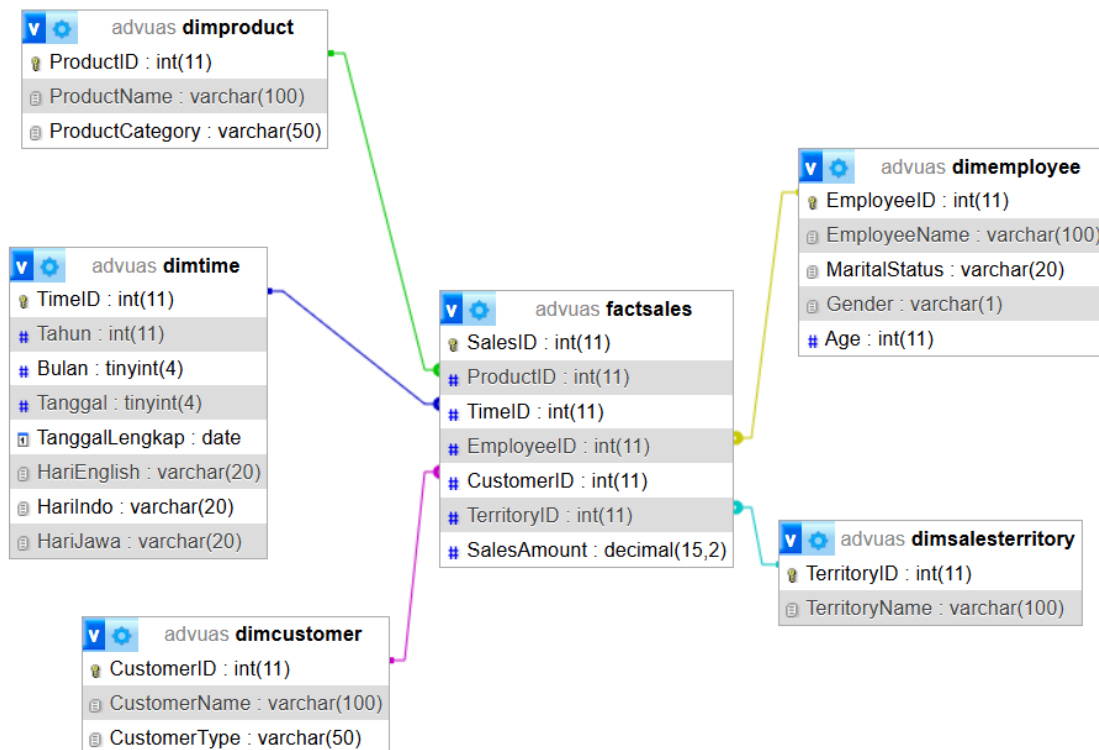
- **Xampp** adalah sebuah perangkat lunak yang berfungsi sebagai server lokal untuk pengembangan website. Nama XAMPP merupakan singkatan dari Cross-platform (X), Apache (A), MySQL/MariaDB (M), PHP (P), dan Perl (P). Tools ini menyediakan lingkungan pengembangan web lokal di mana pengembang dapat membuat, menguji, dan menjalankan aplikasi berbasis PHP dan database MySQL/MariaDB secara offline sebelum mengunggahnya ke server online. Dengan XAMPP, server Apache dan layanan database dapat berjalan secara simultan sehingga sangat memudahkan dalam membangun dan mengembangkan aplikasi web.
- **Pentaho** adalah perangkat lunak yang digunakan dalam solusi Business Intelligence (BI) dan pengolahan data, terutama untuk proses ETL (Extract, Transform, Load). Pentaho membantu dalam mengambil data dari berbagai sumber, mengolah atau mengubah data menjadi format yang diinginkan, dan memuatnya ke dalam database atau data warehouse. Selain itu, Pentaho memungkinkan pembuatan laporan bisnis, dashboard visual, serta analisis data untuk mendukung pengambilan keputusan berbasis data. Tools ini sering digunakan oleh perusahaan untuk mengintegrasikan data dan menyediakan informasi yang relevan dalam bentuk yang mudah dipahami.
- **Visual Studio Code** atau sering disebut VS Code adalah editor kode sumber yang dikembangkan oleh Microsoft. VS Code dikenal sebagai editor kode yang ringan, cepat, dan mendukung berbagai bahasa pemrograman seperti Python, JavaScript, PHP, Java, dan banyak lagi. Selain itu, VS Code dilengkapi dengan fitur-fitur canggih seperti intellisense (autocompletion dan suggestion), debugging, serta integrasi Git. Dengan dukungan ekstensi yang beragam, pengguna dapat menambahkan fungsionalitas sesuai kebutuhan mereka, seperti linting, integrasi tools eksternal, dan manajemen proyek, menjadikannya alat yang ideal untuk pengembangan aplikasi web, desktop, maupun mobile.

## BAB 3

### HASIL & PEMBAHASAN

#### 3.1 Implementasi Multidimensi

Setelah menentukan tabel dimensi, peneliti dapat melakukan implementasi dari data yang telah didapat, yang dimana ditemukan nya 5 dimensi. Lima dimensi ini dapat menjawab dari 4W 1H dan dapat ditampilkan pada dashboard yang akan kita buat. Berikut adalah model multidimensi yang telah dirancang:



Gambar 3.1 Tabel Multidimensi

## Penjelasan Struktur Database advuas

Database advuas adalah basis data yang dirancang untuk analisis data penjualan, yang terdiri dari beberapa tabel dimensi (dimensional tables) dan tabel fakta (fact table). Tabel-tabel ini menggunakan data dari database AdventureWork2019 yang merupakan database contoh.

### 1. Tabel Dimensi (Dimension Tables)

Tabel dimensi menyimpan atribut deskriptif untuk analisis.

#### a. Tabel dimproduct

```
-- Membuat tabel dimproduct

CREATE TABLE dimproduct (

    ProductID INT PRIMARY KEY AUTO_INCREMENT,

    ProductName VARCHAR(100) NOT NULL,

    ProductCategory VARCHAR(50)

);
```

#### b. Table dimcustomer

```
-- Membuat tabel dimcustomer

CREATE TABLE dimcustomer (

    CustomerID INT PRIMARY KEY AUTO_INCREMENT,

    CustomerName VARCHAR(100) NOT NULL,

    CustomerType VARCHAR(50) NOT NULL

);
```

### **c. Table dimemployee**

```
-- Membuat tabel dimemployee

CREATE TABLE dimemployee (

    EmployeeID INT PRIMARY KEY AUTO_INCREMENT,

    EmployeeName VARCHAR(100) NOT NULL,

    MaritalStatus VARCHAR(20),

    Gender VARCHAR(1) DEFAULT NULL,

    Age INT CHECK (Age >= 18)

);
```

### **d. Table dimsalesterritory**

```
-- Membuat tabel dimsalesterritory

CREATE TABLE dimsalesterritory (

    TerritoryID INT PRIMARY KEY AUTO_INCREMENT,

    TerritoryName VARCHAR(100) NOT NULL

);
```

### **e. Table dimtime**

```
-- Membuat tabel dimtime

CREATE TABLE dimtime (

    TimeID INT PRIMARY KEY AUTO_INCREMENT,

    Tahun INT NOT NULL,

    Bulan TINYINT NOT NULL,

    Tanggal TINYINT NOT NULL,

    TanggalLengkap DATE NOT NULL,
```

```

    HariEnglish VARCHAR(20),

    HariIndo VARCHAR(20),

    HariJawa VARCHAR(20)

);

```

### 3.2 Analisis kebutuhan data tabel dimensi

- dimproduct

Data yang di butuhkan	Sumber Data	Keterangan
ProductID	p.ProductID	<p><b>ProductID</b> adalah kolom yang menyimpan ID unik untuk setiap produk.</p> <p>Dalam query ini, <b>ProductID</b> diambil dari tabel <b>production_product</b> yang ada di database <b>AdventureWork2019</b>.</p> <p>ID produk ini digunakan untuk mengidentifikasi setiap produk secara unik dalam sistem.</p> <p><b>ProductID</b> adalah kunci utama di dalam tabel <b>dimproduct</b>, yang mana setiap produk memiliki ID yang berbeda.</p>
ProductName	p.Name AS ProductName	<p><b>ProductName</b> adalah kolom yang menyimpan nama produk.</p> <p>Nama produk diambil dari kolom <b>Name</b> di tabel <b>production_product</b> pada database <b>AdventureWork2019</b>.</p> <p>Nama produk ini digunakan untuk mengenali atau menampilkan nama produk secara lebih jelas dalam</p>

		<p>laporan atau analisis.</p> <p>Dalam query, nama <b>Name</b> dari tabel <b>production_product</b> diubah aliasnya menjadi <b>ProductName</b> untuk disesuaikan dengan kolom dalam tabel <b>dimproduct</b>.</p>
ProductCategory	c.Name AS ProductCategory	<p><b>ProductCategory</b> adalah kolom yang menyimpan kategori produk.</p> <p>Kategori produk ini diambil dari kolom <b>Name</b> di tabel <b>production_productcategory</b> yang ada di database <b>AdventureWork2019</b>.</p> <p>Query ini menggunakan dua <i>join</i> untuk mengambil kategori produk:</p> <ol style="list-style-type: none"> <li>1. <b>LEFT JOIN</b>  <b>production_products</b>  <b>ubcategory s ON</b>  <b>p.ProductSubcategoryID =</b>  <b>s.ProductSubcategoryID:</b> Menggabungkan data produk dengan subkategori produk.</li> <li>2. <b>LEFT JOIN</b>  <b>production_productcategory c ON</b>  <b>s.ProductCategoryID =</b>  <b>c.ProductCategoryID:</b> Menggabungkan data subkategori dengan kategori produk yang lebih besar.</li> </ol> <p>Kategori ini menggambarkan jenis atau klasifikasi dari produk, misalnya elektronik, pakaian, atau aksesoris.</p> <p>Dalam query ini, nama</p>

		kategori produk diubah aliasnya menjadi <b>ProductCategory</b> agar sesuai dengan kolom dalam tabel <b>dimproduct</b> .
--	--	---

#### Query

```

INSERT INTO advuas.dimproduct (ProductID, ProductName,
ProductCategory)
SELECT
    p.ProductID,
    p.Name AS ProductName,
    c.Name AS ProductCategory
FROM
    AdventureWork2019.production_product p
LEFT JOIN
    AdventureWork2019.production_productsubcategory s
    ON p.ProductSubcategoryID = s.ProductSubcategoryID
LEFT JOIN
    AdventureWork2019.production_productcategory c
    ON s.ProductCategoryID = c.ProductCategoryID;

```

#### • dimcustomer

Data yang di butuhkan	Sumber data	Keterangan
CustomerID	c.CustomerID	<b>CustomerID</b> adalah kolom yang menyimpan ID unik untuk setiap customer (pelanggan). Kolom <b>CustomerID</b> diambil langsung dari tabel <b>sales_customer</b> yang ada di database <b>AdventureWork2019</b> . <b>CustomerID</b> digunakan untuk mengidentifikasi



Data yang di butuhkan	Sumber data	Keterangan
		secara unik setiap pelanggan dalam tabel <b>dimcustomer</b> . <b>CustomerID</b> di sini adalah nilai yang ditarik dari tabel <b>sales_customer</b> yang berisi informasi dasar mengenai setiap pelanggan, seperti ID dan keterkaitannya dengan entitas bisnis.
CustomerName	CONCAT_WS(' ', p.FirstName, p.MiddleName, p.LastName) AS CustomerName	<b>CustomerName</b> adalah kolom yang menyimpan nama lengkap pelanggan. Nama lengkap pelanggan dihasilkan dengan menggabungkan tiga kolom dari tabel <b>person_person</b> yaitu <b>FirstName</b> , <b>MiddleName</b> , dan <b>LastName</b> . Fungsi <b>CONCAT_WS(' ', ...)</b> digunakan untuk menggabungkan nama depan ( <b>FirstName</b> ), nama tengah ( <b>MiddleName</b> ), dan nama belakang ( <b>LastName</b> ) dengan spasi sebagai pemisah antara nama-nama tersebut. Hasil gabungan ini menjadi nama lengkap pelanggan yang akan disimpan dalam kolom <b>CustomerName</b> pada tabel <b>dimcustomer</b> .
CustomerType	p.PersonType AS CustomerType	CustomerType adalah kolom yang menyimpan jenis atau tipe pelanggan. PersonType diambil dari tabel <b>person_person</b> yang ada di database AdventureWork2019.

Data yang di butuhkan	Sumber data	Keterangan
		<p>PersonType mengindikasikan kategori pelanggan, apakah pelanggan individu (seperti IN untuk Individual) atau perusahaan (seperti OR untuk Organization).</p> <p>CustomerType yang dihasilkan di sini akan mengklasifikasikan setiap pelanggan ke dalam tipe tertentu, yang memudahkan analisis berdasarkan jenis pelanggan.</p>

## Query

```

INSERT INTO advuas.dimcustomer (CustomerID, CustomerName,
CustomerType)
SELECT
    c.CustomerID,
    CONCAT_WS(' ', p.FirstName, p.MiddleName, p.LastName)
AS CustomerName,
    p.PersonType AS CustomerType
FROM
    AdventureWork2019.sales_customer c
JOIN
    AdventureWork2019.person_person p
ON
    c.PersonID = p.BusinessEntityID;

```

- dimemployee

Data yang dibutuhkan	Sumber Data	Keterangan
EmployeeID	p.BusinessEntityID AS EmployeeID	<p><b>EmployeeID</b> adalah kolom yang menyimpan ID unik untuk setiap pegawai.</p> <p>Data <b>EmployeeID</b> diambil dari kolom <b>BusinessEntityID</b> yang ada di tabel <b>person_person</b>.</p> <p><b>BusinessEntityID</b> merupakan ID unik yang digunakan untuk mengidentifikasi entitas bisnis yang bersangkutan dalam sistem perusahaan (dalam hal ini, pegawai).</p> <p><b>EmployeeID</b> akan menjadi nilai yang mengidentifikasi pegawai secara unik dalam tabel <b>dimemployee</b>.</p>

EmployeeName	CONCAT_WS(' ', p.FirstName, p.MiddleName, p.LastName) AS EmployeeName	<b>EmployeeName</b> adalah kolom yang menyimpan nama lengkap pegawai. Nama lengkap pegawai diperoleh dengan menggabungkan kolom <b>FirstName</b> , <b>MiddleName</b> , dan <b>LastName</b> yang ada di tabel <b>person_person</b> menggunakan fungsi <b>CONCAT_WS(' ', ...)</b> . Fungsi <b>CONCAT_WS</b> menggabungkan nilai-nilai ini dengan spasi sebagai pemisah antar bagian nama. Dengan cara ini, kita mendapatkan nama lengkap pegawai, yang kemudian akan disimpan dalam kolom <b>EmployeeName</b> pada tabel <b>dimemployee</b> .
MaritalStatus	h.MaritalStatus	<b>MaritalStatus</b> adalah kolom yang menyimpan status pernikahan pegawai. Data <b>MaritalStatus</b> diambil dari kolom <b>MaritalStatus</b> yang ada di tabel <b>humanresources_employe</b> . Kolom ini berisi informasi apakah pegawai tersebut sudah menikah atau belum, misalnya "M" untuk menikah, "S" untuk belum menikah, atau status lain yang digunakan oleh sistem.
Gender	h.Gender	<b>Gender</b> adalah kolom yang menyimpan jenis kelamin

		<p>pegawai.</p> <p>Data <b>Gender</b> diambil dari kolom <b>Gender</b> yang ada di tabel <b>humanresources_employe</b>.</p> <p>Kolom ini menunjukkan jenis kelamin pegawai, yang umumnya diwakili dengan "M" untuk laki-laki atau "F" untuk perempuan. Data ini akan disalin ke kolom <b>Gender</b> dalam tabel <b>dimemployee</b>.</p>
Age	<p><math>\text{YEAR('2024-01-01')} - \text{YEAR(h.BirthDate)}</math> AS Age</p>	<p><b>Age</b> adalah kolom yang menyimpan usia pegawai. Usia dihitung dengan mengurangi tahun kelahiran pegawai (<b>BirthDate</b>) yang ada di tabel <b>humanresources_employe</b> dari tahun 2024. Fungsi <b>YEAR()</b> digunakan untuk mengambil tahun dari tanggal kelahiran.</p> <p><b>Age</b> dihitung dengan cara: <math>2024 - \text{YEAR(BirthDate)}</math>. Ini akan memberikan usia pegawai pada tanggal 1 Januari 2024, yang kemudian disalin ke kolom <b>Age</b> pada tabel <b>dimemployee</b>.</p> <p>Data <b>BirthDate</b> diambil dari kolom <b>BirthDate</b> di tabel <b>humanresources_employe</b>.</p>

## Query

```
INSERT INTO advuas.dimemployee (EmployeeID, EmployeeName,
MaritalStatus, Gender, Age)
SELECT
    p.BusinessEntityID AS EmployeeID,
    CONCAT_WS(' ', p.FirstName, p.MiddleName, p.LastName)
AS EmployeeName,
    h.MaritalStatus,
    h.Gender,
    YEAR('2024-01-01') - YEAR(h.BirthDate) AS Age
FROM
    AdventureWork2019.person_person p
JOIN
    AdventureWork2019.humanresources_employee h
ON
    p.BusinessEntityID = h.BusinessEntityID
WHERE
    p.PersonType = 'SP'
```

- dimsalesterritory

Data yang di butuhkan	Sumber Data	Keterangan
TerritoryID	t.TerritoryID	<b>TerritoryID</b> adalah kolom yang menyimpan ID unik untuk setiap wilayah penjualan (territory). Data <b>TerritoryID</b> diambil dari kolom <b>TerritoryID</b> yang ada di tabel <b>sales_salesterritory</b> pada database <b>AdventureWork2019</b> . Kolom <b>TerritoryID</b> berfungsi sebagai pengenal atau ID unik untuk setiap wilayah yang terkait dengan penjualan dalam sistem. <b>TerritoryID</b> akan dimasukkan ke dalam

		kolom <b>TerritoryID</b> di tabel <b>dimsalesterritory</b> pada database <b>advuas</b> untuk menyimpan referensi ke wilayah penjualan di data warehouse
TerritoryName	t.Name AS TerritoryName	<b>TerritoryName</b> adalah kolom yang menyimpan nama wilayah penjualan. Data <b>TerritoryName</b> diambil dari kolom <b>Name</b> yang ada di tabel <b>sales_salesterritory</b> pada database <b>AdventureWork2019</b> . Kolom <b>Name</b> berisi nama deskriptif dari wilayah penjualan, yang menggambarkan area geografis atau area tanggung jawab penjualan tertentu dalam perusahaan. Data <b>TerritoryName</b> akan disalin ke dalam kolom <b>TerritoryName</b> pada tabel <b>dimsalesterritory</b> dalam data warehouse <b>advuas</b> .

#### Query

```

dimsalesterritory

INSERT INTO advuas.dimsalesterritory (TerritoryID,
TerritoryName)
SELECT
    t.TerritoryID,
    t.Name AS TerritoryName
FROM
    adventurework2019.sales_salesterritory t;

```

- dimtime

Prosedur **isiDimTime** yang dibuat menggunakan MySQL ini bertujuan untuk mengisi tabel **dimtime** dengan data waktu dalam periode tertentu, mulai dari tanggal **waktuMulai** hingga **waktuSelesai**. Prosedur ini menerima tiga parameter: **waktuMulai** (tanggal mulai), **waktuSelesai** (tanggal akhir), dan **hariPasarPertama** (hari pasar pertama yang dimulai dengan salah satu dari lima jenis hari pasar Jawa: 'legi', 'pahing', 'pon', 'wage', atau 'kliwon').

Dalam prosedur ini, sebuah **WHILE loop** digunakan untuk melakukan iterasi mulai dari **waktuMulai** hingga **waktuSelesai**. Pada setiap iterasi, data tentang tanggal tersebut disisipkan ke dalam tabel **dimtime**. Kolom yang diisi antara lain:

- **tahun**: Tahun dari tanggal,
- **bulan**: Bulan dari tanggal,
- **tanggal**: Tanggal dari tanggal,
- **tanggallengkap**: Tanggal lengkap,
- **harienglish**: Nama hari dalam bahasa Inggris (misalnya, Senin, Selasa),
- **hariindo**: Nama hari dalam bahasa Indonesia (misalnya, Minggu, Senin),
- **harijawa**: Hari pasar Jawa, yang dimulai dari nilai parameter **hariPasarPertama** dan berubah sesuai urutan hari pasar (legi, pahing, pon, wage, kliwon).

Setelah setiap insert, tanggal **waktuMulai** diperbarui dengan menambahkan satu hari, dan **hariPasarSaatIni** diperbarui dengan mengikuti siklus urutan hari pasar yang telah ditentukan (legi -> pahing -> pon -> wage -> kliwon -> legi kembali).

Prosedur ini akan terus berlanjut hingga mencapai tanggal **waktuSelesai**. Dengan cara ini, tabel **dimtime** akan terisi dengan data tanggal dan informasi terkait waktunya

Query

```
DELIMITER $$

CREATE PROCEDURE isiDimTime(
    IN waktuMulai DATE,
    IN waktuSelesai DATE,
    IN hariPasarPertama ENUM('legi', 'pahing', 'pon', 'wage',
    'kliwon')
```



```

)
BEGIN
    DECLARE hariPasarSaatIni ENUM('legi', 'pahing', 'pon',
    'wage', 'kliwon') DEFAULT hariPasarPertama;

    WHILE waktuMulai <= waktuSelesai DO
        INSERT INTO dimtime (tahun, bulan, tanggal,
    tanggallengkap, harienglish, hariindo, harijawa)
        VALUES (
            YEAR(waktuMulai), -- Tahun dari tanggal
            MONTH(waktuMulai), -- Bulan dari tanggal
            DAY(waktuMulai), -- Tanggal dari tanggal
            waktuMulai, -- Tanggal lengkap
            DAYNAME(waktuMulai), -- Nama hari dalam bahasa
Inggris
            CASE DAYOFWEEK(waktuMulai) -- Nama hari dalam
bahasa Indonesia
                WHEN 1 THEN 'Minggu'
                WHEN 2 THEN 'Senin'
                WHEN 3 THEN 'Selasa'
                WHEN 4 THEN 'Rabu'
                WHEN 5 THEN 'Kamis'
                WHEN 6 THEN 'Jumat'
                WHEN 7 THEN 'Sabtu'
            END,
            hariPasarSaatIni -- Hari pasar Jawa saat ini
        );

        -- Tambah satu hari ke waktuMulai
        SET waktuMulai = DATE_ADD(waktuMulai, INTERVAL 1 DAY);

        -- Perbarui hari pasar
        SET hariPasarSaatIni = CASE
            WHEN hariPasarSaatIni = 'legi' THEN 'pahing'
            WHEN hariPasarSaatIni = 'pahing' THEN 'pon'
            WHEN hariPasarSaatIni = 'pon' THEN 'wage'
            WHEN hariPasarSaatIni = 'wage' THEN 'kliwon'
            WHEN hariPasarSaatIni = 'kliwon' THEN 'legi'
        END;
    END WHILE;
END$$

DELIMITER ;

```

NB :gunakan “CALL isiDimTime('2023-01-01', '2023-01-31', 'legi');” untuk mengisi tabel

### 3.3 Tabel Fakta

-- Membuat tabel factsales

```
CREATE TABLE factsales (  
  
    SalesID INT PRIMARY KEY AUTO_INCREMENT,  
  
    ProductID INT NOT NULL,  
  
    TimeID INT NOT NULL,  
  
    EmployeeID INT NOT NULL,  
  
    CustomerID INT NOT NULL,  
  
    TerritoryID INT NOT NULL,  
  
    SalesAmount DECIMAL(15, 2) NOT NULL,  
  
    FOREIGN KEY (ProductID) REFERENCES  
dimproduct(ProductID),  
  
    FOREIGN KEY (TimeID) REFERENCES dimtime(TimeID),  
  
    FOREIGN KEY (EmployeeID) REFERENCES  
dimemployee(EmployeeID),  
  
    FOREIGN KEY (CustomerID) REFERENCES  
dimcustomer(CustomerID),  
  
    FOREIGN KEY (TerritoryID) REFERENCES  
dimsalesterritory(TerritoryID)  
  
);
```

### 3.4 Relevansi Tabel fakta dengan tabel dimensi

- **Tabel Fakta (factsales):**
    - Tabel ini menyimpan data transaksi penjualan secara detail, seperti jumlah penjualan (*SalesAmount*) dan hubungan ke dimensi lain menggunakan *foreign key*.
    - Setiap baris dalam tabel ini mewakili sebuah transaksi.
  - **Tabel Dimensi:**
    - **dimcustomer:** Berisi informasi pelanggan, seperti ID, nama, dan tipe pelanggan.
    - **dimemployee:** Berisi informasi karyawan, seperti ID, nama, status pernikahan, jenis kelamin, dan usia.
    - **dimproduct:** Berisi detail produk, seperti ID, nama produk, dan kategori produk.
    - **dimsalesterritory:** Berisi wilayah atau area penjualan.
    - **dimtime:** Berisi informasi waktu dengan granularity hingga harian, termasuk hari pasar Jawa.
  - Tabel dimensi mendukung analisis mendalam dengan memberikan konteks pada data transaksi di tabel fakta.
  - **Relevansi Antar Tabel:**
    - Tabel fakta terhubung dengan tabel dimensi melalui kolom *foreign key* (misalnya, **ProductID**, **TimeID**, dll.), sehingga memungkinkan pengelompokan data transaksi berdasarkan atribut dimensi tertentu.
    - Contoh hubungan:
      - *factsales.ProductID* → *dimproduct.ProductID* → Analisis berdasarkan produk.
      - *factsales.TimeID* → *dimtime.TimeID* → Analisis berdasarkan waktu.
      - *factsales.EmployeeID* → *dimemployee.EmployeeID* → Analisis berdasarkan karyawan.
      - *factsales.CustomerID* → *dimcustomer.CustomerID* → Analisis berdasarkan pelanggan.
      - *factsales.TerritoryID* → *dimsalesterritory.TerritoryID* → Analisis berdasarkan wilayah.
- 

#### Output atau Analisis yang Dapat Dilakukan:

1. **Analisis Penjualan Berdasarkan Waktu:**
  - Berapa total penjualan bulanan pada tahun tertentu?
  - Apa hari terbaik untuk penjualan (misalnya, hari pasar Jawa seperti *kliwon* vs *legi*)?
2. **Analisis Pelanggan:**
  - Kategori pelanggan mana yang menyumbang penjualan terbesar (misalnya, pelanggan individu vs perusahaan)?
  - Identifikasi pelanggan dengan kontribusi penjualan tertinggi.
3. **Analisis Kinerja Karyawan:**
  - Karyawan mana yang memiliki performa penjualan terbaik?

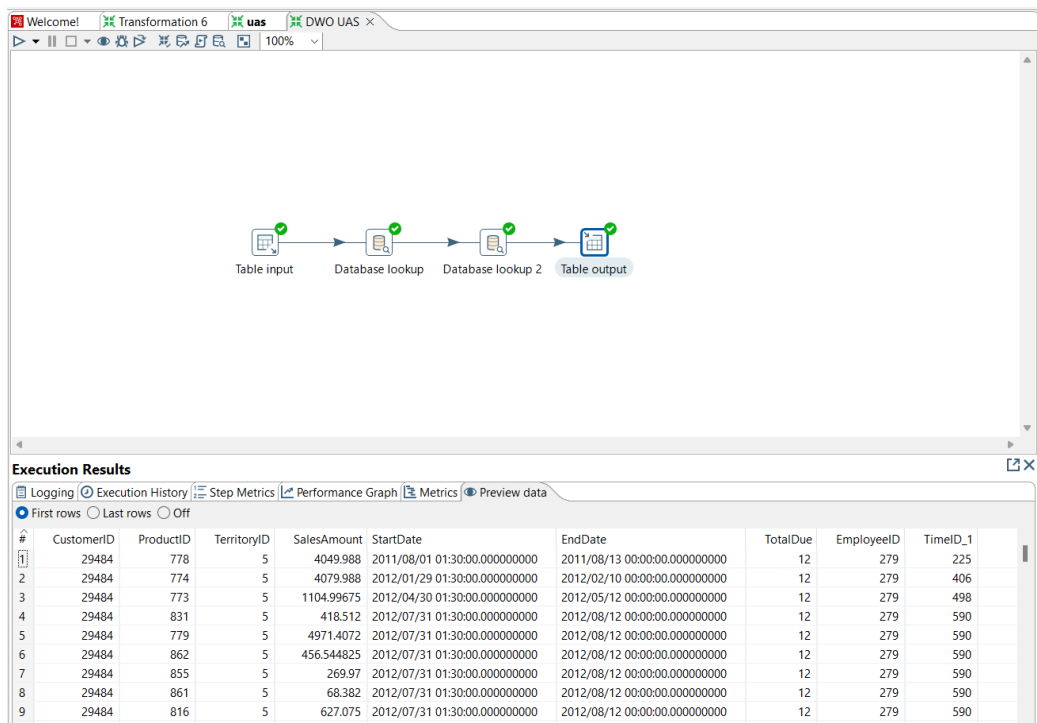
- Bagaimana distribusi penjualan berdasarkan status pernikahan atau usia karyawan?
- 4. **Analisis Produk:**
  - Produk mana yang paling banyak terjual?
  - Bagaimana penjualan suatu kategori produk dibandingkan dengan kategori lainnya?
- 5. **Analisis Wilayah Penjualan:**
  - Wilayah mana yang memberikan kontribusi penjualan terbesar?
  - Apakah ada pola penjualan yang spesifik di wilayah tertentu?
- 6. **Analisis Gabungan:**
  - Bagaimana kontribusi masing-masing karyawan di setiap wilayah penjualan?
  - Analisis penjualan produk tertentu oleh pelanggan tertentu pada periode tertentu.

### 3.5 Proses ETL & OLAP

Pada tahap ini dilakukan Data Integration dengan menggunakan tools Pentaho, untuk penjelasan dan langkah- langkah sebagai berikut :

#### Proses ETL &Olap untuk tabel FactSales

1. Pertama buat file transformation baru dan berikan nama sesuai kebutuhan.
2. Setelah itu buat design sesuai kebutuhan, dapat dilihat pada gambar berikut :

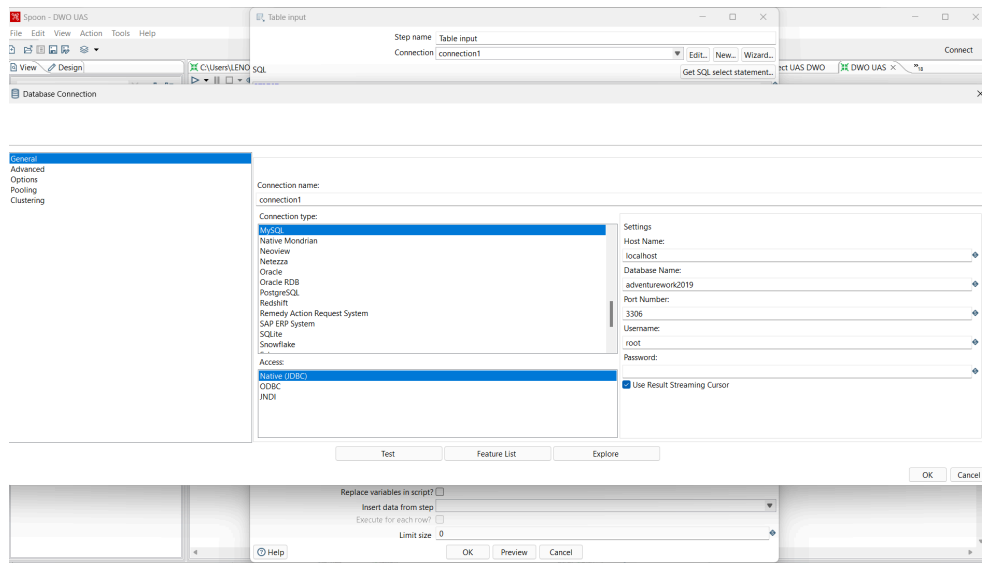


**Execution Results**

#	CustomerID	ProductID	TerritoryID	SalesAmount	StartDate	EndDate	TotalDue	EmployeeID	TimeID_1
1	29484	778	5	4049.988	2011/08/01 01:30:00.000000000	2011/08/13 00:00:00.000000000	12	279	225
2	29484	774	5	4079.988	2012/01/29 01:30:00.000000000	2012/02/10 00:00:00.000000000	12	279	406
3	29484	773	5	1104.99675	2012/04/30 01:30:00.000000000	2012/05/12 00:00:00.000000000	12	279	498
4	29484	831	5	418.512	2012/07/31 01:30:00.000000000	2012/08/12 00:00:00.000000000	12	279	590
5	29484	779	5	4971.4072	2012/07/31 01:30:00.000000000	2012/08/12 00:00:00.000000000	12	279	590
6	29484	862	5	456.544825	2012/07/31 01:30:00.000000000	2012/08/12 00:00:00.000000000	12	279	590
7	29484	855	5	269.97	2012/07/31 01:30:00.000000000	2012/08/12 00:00:00.000000000	12	279	590
8	29484	861	5	68.382	2012/07/31 01:30:00.000000000	2012/08/12 00:00:00.000000000	12	279	590
9	29484	816	5	627.075	2012/07/31 01:30:00.000000000	2012/08/12 00:00:00.000000000	12	279	590

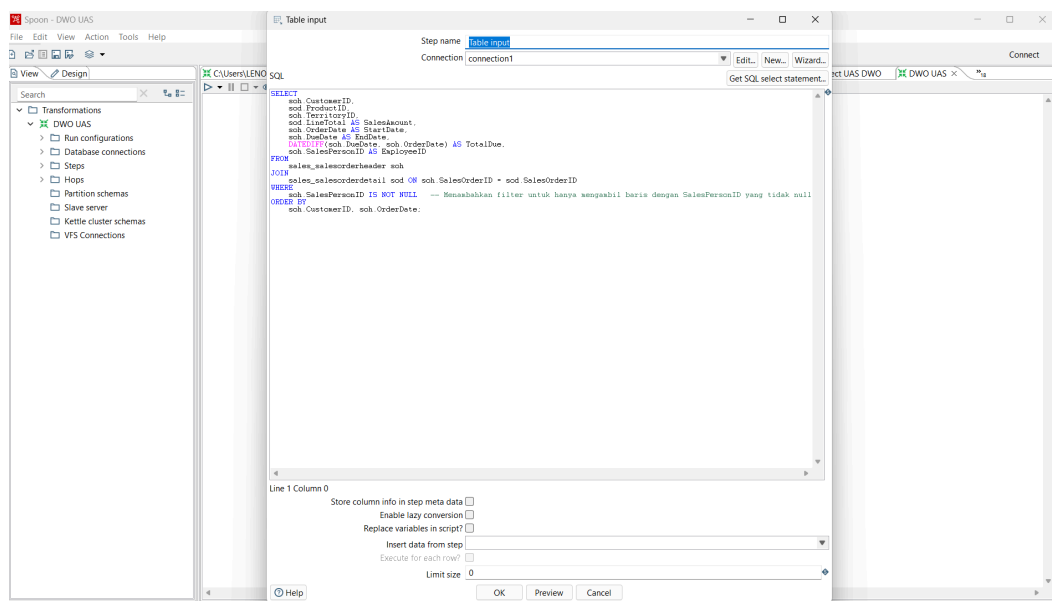
Gambar ETL

1. Doubleclick pada tabel input, lalu klik New
2. Isi koneksi dan sesuaikan kebutuhan, pada kasus ini table input terkoneksi dengan database adventureworks dapat dilihat pada gambar berikut :



Gambar pada table input untuk koneksi ke database

## Tabel Input



berikut SQL untuk SELECT yang ada pada tabel input:

```
SELECT

soh.CustomerID,

sod.ProductID,

soh.TerritoryID,

sod.LineTotal AS SalesAmount,

soh.OrderDate AS StartDate,

soh.DueDate AS EndDate,

DATEDIFF(soh.DueDate, soh.OrderDate) AS TotalDue,

soh.SalesPersonID AS EmployeeID

FROM

sales_salesorderheader soh

JOIN

sales_salesorderdetail sod ON soh.SalesOrderID = sod.SalesOrderID

WHERE

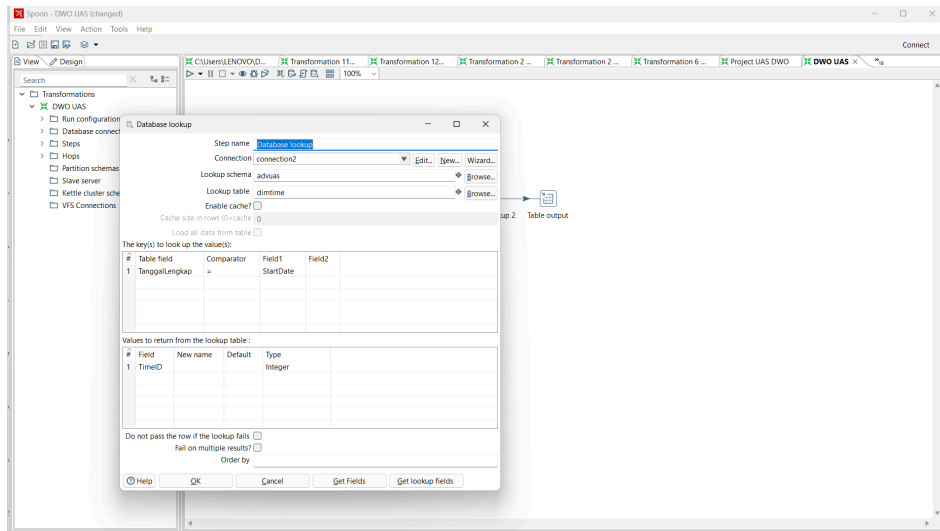
soh.SalesPersonID IS NOT NULL

ORDER BY

soh.CustomerID, soh.OrderDate;
```

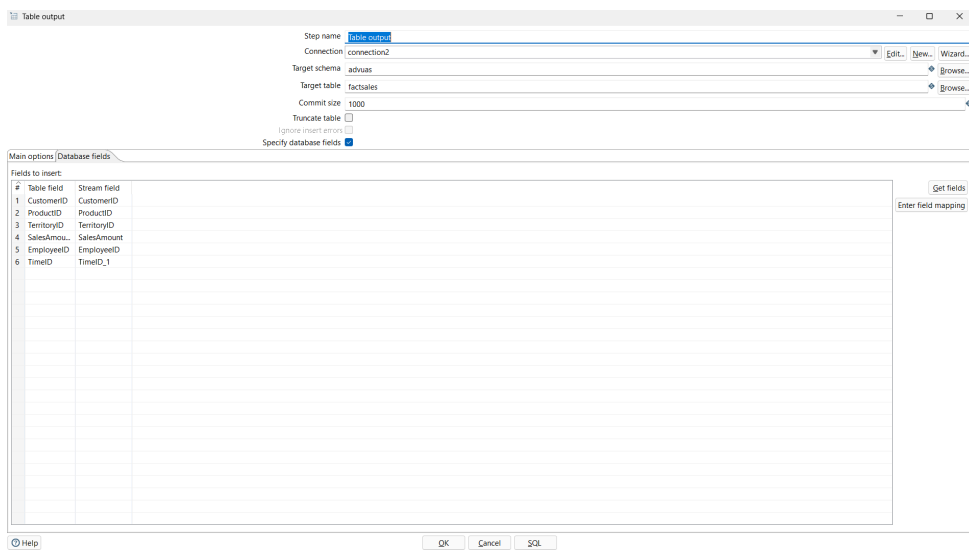
## Database Lookup:

Pada Database lookup 1 bertujuan untuk melakukan lookup TanggalLengkap untuk di lookup berdasarkan TimeID, dapat dilihat seperti gambar berikut:



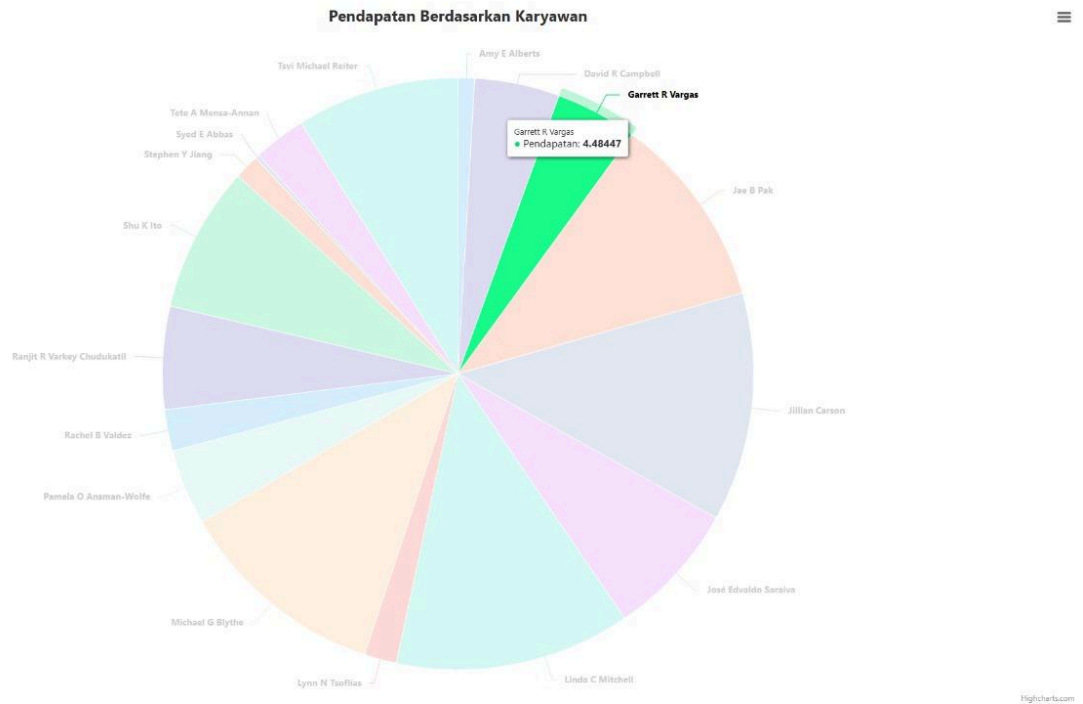
## Table Output:

Pada table output bertujuan untuk memasukan data yang dibutuhkan di tabel factsales, bisa dilihat pada gambar berikut:



### 3.6 Implementasi Dashboard & Hasil Insight Analisis

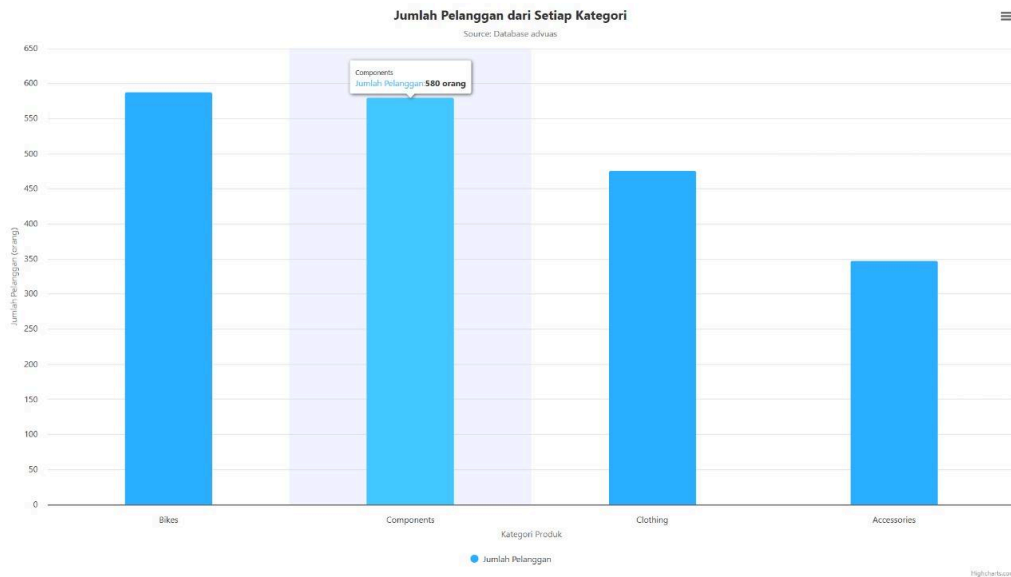
#### 3.6.1 Analisis Pendapatan Hasil Kerja Karyawan



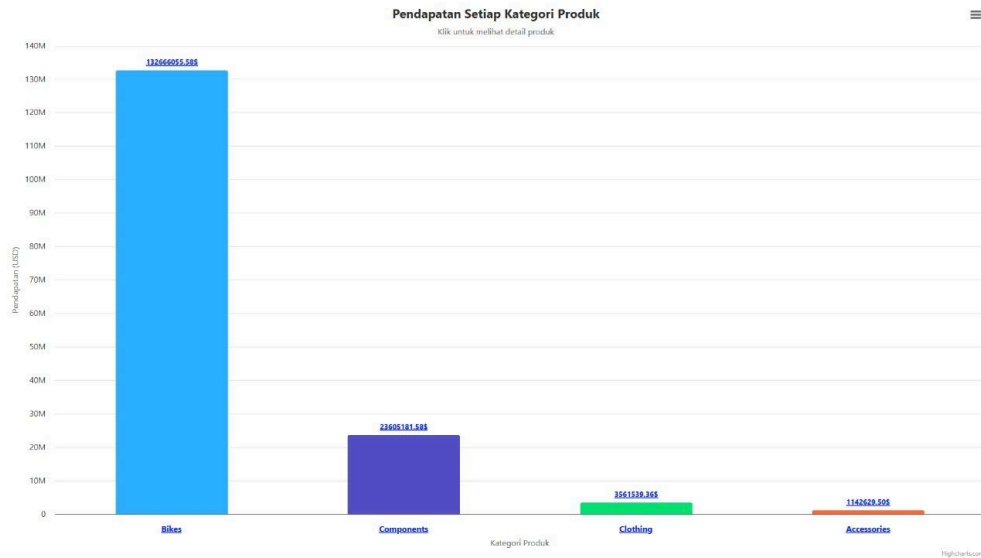
Dari hasil analisis yang dilakukan dan hasil grafik diatas menunjukkan hasil dari penjualan dari karyawan ke pelanggan. data menunjukkan jika Garrett R.Vargas mendapatkan pendapat sebesar 4.48447. dari data tersebut lebih banyak lagi data pendapat karyawan seperti Jae B Pak, Jillian Carson, Jose Edvaldo Saraiva, Linda C. Mitchell, Lynn N Tsofias, Michael G Biythe, Pamela O Ansan-Wolfe, dll



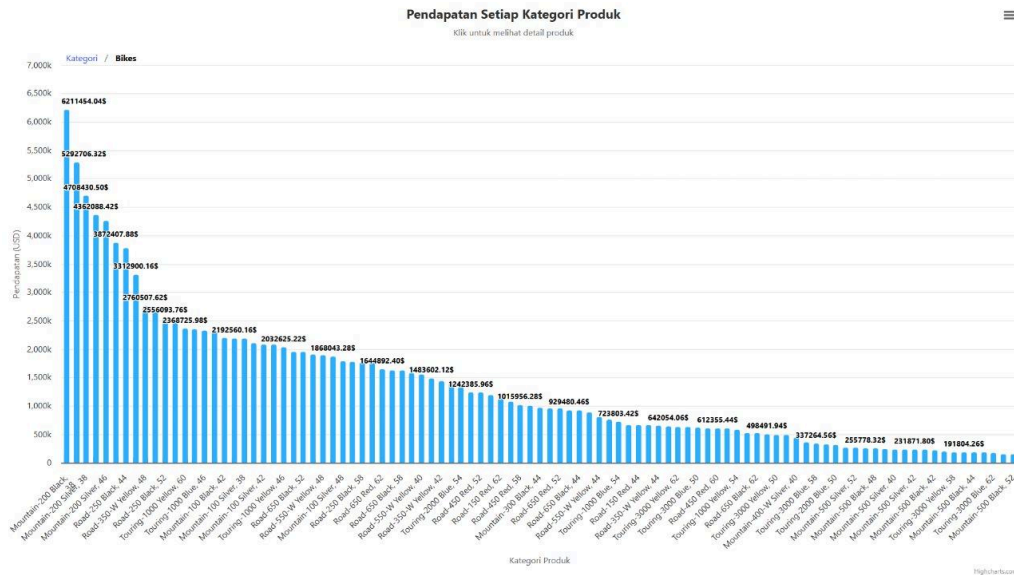
### 3.6.2 Analisis total penjualan berdasarkan kategori



Grafik tersebut menampilkan jumlah pelanggan dari setiap kategori produk dalam bentuk diagram batang. Judul grafik, "Jumlah Pelanggan dari Setiap Kategori," menunjukkan bahwa data yang ditampilkan menggambarkan jumlah pelanggan dari berbagai kategori produk. Pada sumbu horizontal (X), terdapat empat kategori produk, yaitu Bikes, Components, Clothing, dan Accessories. Sedangkan sumbu vertikal (Y) menunjukkan jumlah pelanggan dalam satuan orang. Berdasarkan grafik, kategori Bikes memiliki jumlah pelanggan tertinggi, yaitu sekitar 600 orang, diikuti oleh kategori Components dengan 580 pelanggan. Selanjutnya, kategori Clothing mencatat jumlah pelanggan sekitar 500 orang, dan kategori Accessories berada di posisi terakhir dengan sekitar 400 pelanggan. Grafik ini dilengkapi dengan tooltip yang memberikan informasi jumlah pelanggan secara lebih detail saat cursor diarahkan ke batang tertentu, seperti pada kategori Components yang menampilkan jumlah pelanggan sebanyak 580 orang. Data pada grafik ini diambil dari "Database advuas," sebagaimana dicantumkan di bagian bawah grafik. Keseluruhan data memberikan gambaran bahwa kategori Bikes dan Components menjadi produk unggulan berdasarkan jumlah pelanggan yang tinggi, sedangkan kategori Accessories memiliki performa terendah.



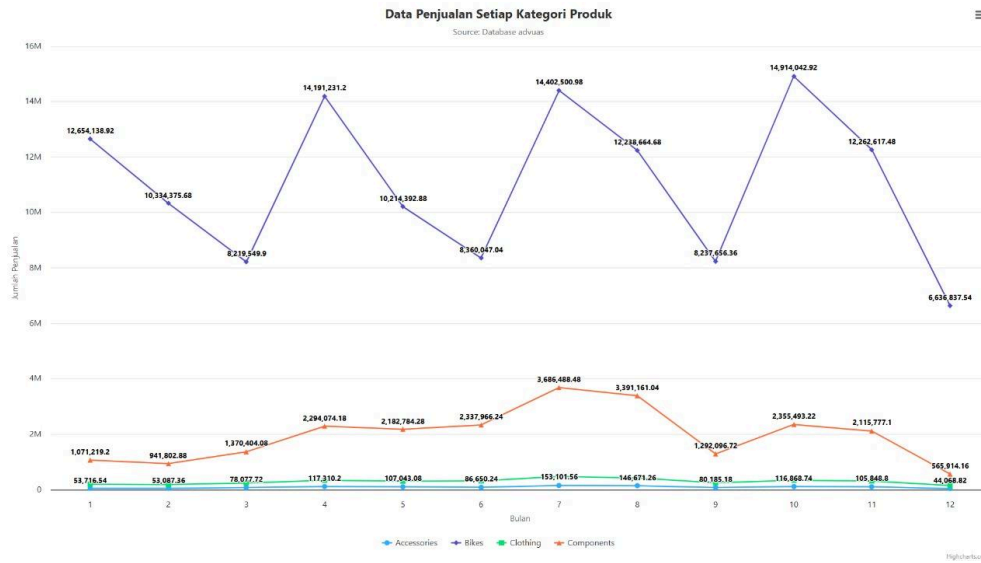
Grafik tersebut menunjukkan pendapatan dari setiap kategori produk, yaitu Bikes, Components, Clothing, dan Accessories, dalam satuan USD. Dari data yang ditampilkan, kategori Bikes mendominasi dengan pendapatan sebesar **132,660,055.59 USD**, jauh melampaui kategori lainnya. Kategori Components berada di posisi kedua dengan pendapatan sebesar **23,605,181.58 USD**, yang masih signifikan tetapi terpaut cukup jauh dari Bikes. Selanjutnya, kategori Clothing mencatat pendapatan sebesar **3,561,539.36 USD**, sedangkan Accessories berada di posisi terakhir dengan pendapatan terendah, yaitu **1,142,629.50 USD**. Grafik ini memperlihatkan ketimpangan yang sangat besar, di mana pendapatan kategori Bikes hampir enam kali lipat lebih besar daripada Components dan jauh melampaui dua kategori lainnya. Hal ini mengindikasikan bahwa Bikes merupakan kategori dengan kontribusi terbesar terhadap total pendapatan perusahaan, kemungkinan disebabkan oleh harga jual yang lebih tinggi serta volume penjualan yang signifikan. Sementara itu, kategori Clothing dan Accessories memberikan kontribusi yang relatif kecil, menunjukkan potensi untuk pengembangan lebih lanjut guna meningkatkan performa mereka. Analisis ini dapat membantu perusahaan menentukan prioritas dalam strategi pemasaran, investasi, dan pengembangan produk di masa mendatang.



Grafik tersebut menampilkan pendapatan dari berbagai produk dalam kategori *Bikes*. Setiap batang pada grafik merepresentasikan pendapatan untuk masing-masing jenis sepeda dengan satuan USD. Produk dengan pendapatan tertinggi adalah *Mountain-200 Black*, yang mencapai **6,211,454.04 USD**, diikuti oleh *Mountain-200 Silver* dengan **5,297,076.32 USD**, dan *Road-550-W Yellow* dengan **4,708,430.50 USD**. Setelah itu, terdapat penurunan pendapatan yang bertahap pada produk-produk lainnya, seperti *Mountain-200 Blue* dan *Touring-1000 Blue*.

Grafik ini menunjukkan bahwa sebagian besar pendapatan berasal dari beberapa produk unggulan, sedangkan sebagian besar produk lainnya memiliki kontribusi pendapatan yang lebih kecil. Produk dengan pendapatan di bawah **1 juta USD** terlihat mendominasi bagian bawah grafik, mencerminkan distribusi yang tidak merata di mana pendapatan utama hanya berasal dari segelintir produk teratas. Hal ini mengindikasikan pentingnya fokus pada produk-produk unggulan untuk memaksimalkan pendapatan, sambil tetap mempertimbangkan strategi peningkatan performa untuk produk dengan pendapatan lebih rendah. Grafik ini memberikan wawasan yang jelas untuk memprioritaskan investasi pada produk dengan performa terbaik dalam kategori *Bikes*.

### 3.6.2 Analisis Data Penjualan Setiap Kategori Produk



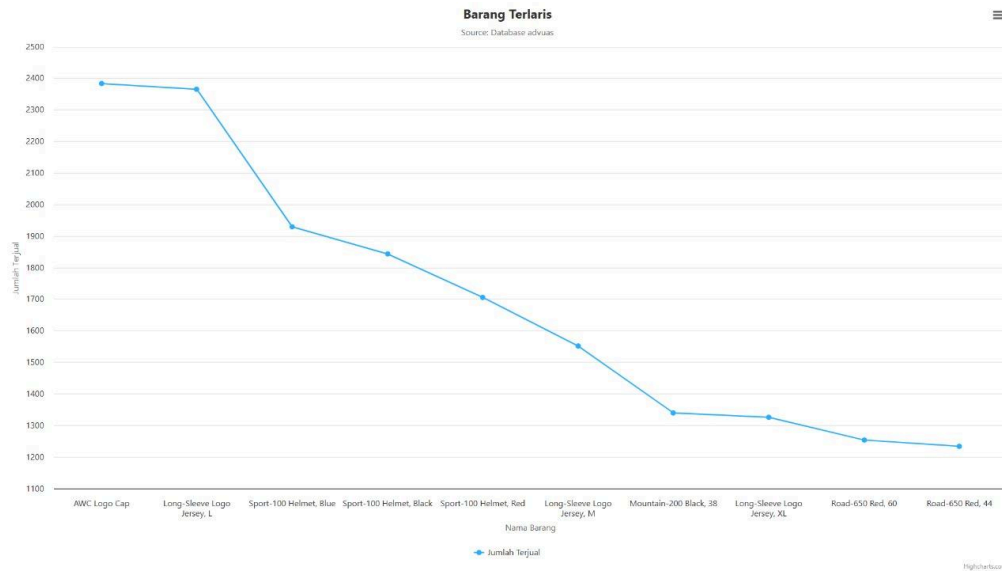
Grafik tersebut menunjukkan data penjualan untuk setiap kategori produk selama 12 bulan. Terdapat empat kategori produk yang ditampilkan, yaitu Accessories, Bikes, Clothing, dan Components. Kategori *Bikes* memiliki penjualan tertinggi sepanjang tahun dengan fluktuasi yang signifikan. Puncak penjualan terjadi pada bulan ke-4 (14,191,231.2), bulan ke-7 (14,402,500.98), dan bulan ke-10 (14,914,042.92). Sebaliknya, penurunan drastis terlihat pada bulan ke-12 dengan penjualan mencapai titik terendah di 6,636,837.54.

Kategori *Components* menempati posisi kedua dengan tren yang relatif stabil namun mengalami lonjakan pada bulan ke-7 dengan penjualan sebesar 3,686,488.48. Penurunan signifikan terlihat pada bulan ke-12 dengan angka penjualan hanya 565,914.16.

Kategori *Clothing* dan *Accessories* memiliki penjualan yang jauh lebih rendah dibanding dua kategori lainnya. Penjualan *Clothing* cenderung stabil sepanjang tahun dengan sedikit peningkatan pada bulan ke-8 (146,671.26), sedangkan *Accessories* menunjukkan pola yang serupa dengan penjualan tertinggi di bulan ke-7 (153,101.56) dan mengalami penurunan pada bulan ke-12 (44,068.82).

Secara keseluruhan, kategori *Bikes* mendominasi penjualan dengan variasi yang cukup besar, sementara kategori lainnya menunjukkan tren yang lebih stabil namun dengan kontribusi yang jauh lebih kecil terhadap total penjualan.

### 3.6.3 Analisis Data Barang Terlaris

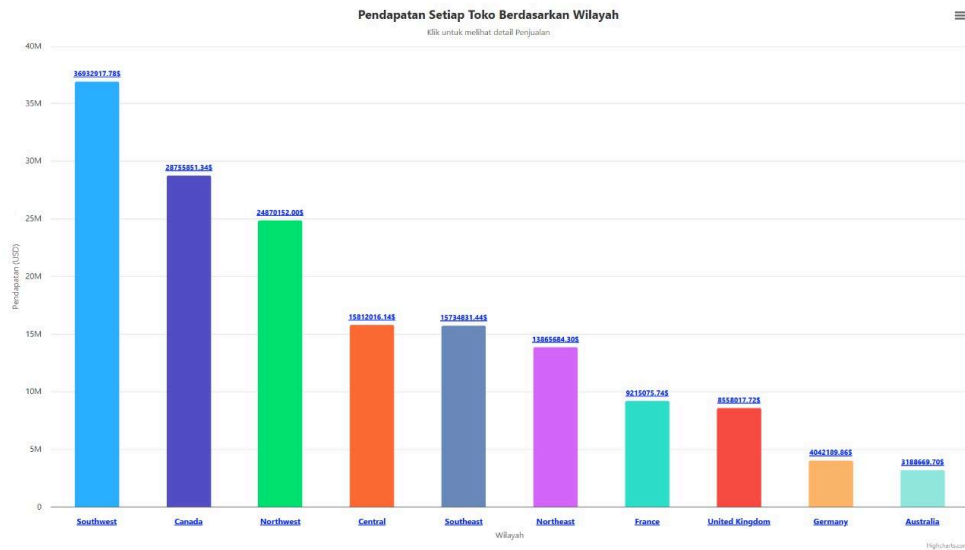


Grafik tersebut menunjukkan daftar barang terlaris berdasarkan jumlah unit yang terjual. Produk dengan penjualan tertinggi adalah **AWC Logo Cap** dan **Long-Sleeve Logo Jersey, L**, dengan jumlah penjualan yang hampir sama, yaitu sekitar 2.400 unit. Setelah itu, terjadi penurunan signifikan pada produk **Sport-100 Helmet, Blue**, diikuti oleh **Sport-100 Helmet, Black** dan **Sport-100 Helmet, Red**, dengan jumlah penjualan berkisar antara 1.900 hingga 1.700 unit.

Produk **Long-Sleeve Logo Jersey, M** dan **Mountain-200 Black, 38** menunjukkan tren penurunan lebih lanjut dengan penjualan sekitar 1.600 dan 1.400 unit. Di bagian bawah grafik, produk dengan penjualan paling rendah adalah **Long-Sleeve Logo Jersey, XL**, **Road-650 Red, 60**, dan **Road-650 Red, 44**, dengan jumlah penjualan berkisar antara 1.300 hingga 1.200 unit.

Secara keseluruhan, grafik ini memperlihatkan bahwa produk pakaian seperti jersey dan helm mendominasi daftar barang terlaris, dengan beberapa produk sepeda juga muncul di posisi bawah daftar. Penjualan cenderung menurun secara bertahap dari produk teratas hingga produk dengan penjualan terendah.

### 3.6.3 Analisis Pendapatan Setiap Toko Berdasarkan Wilayah

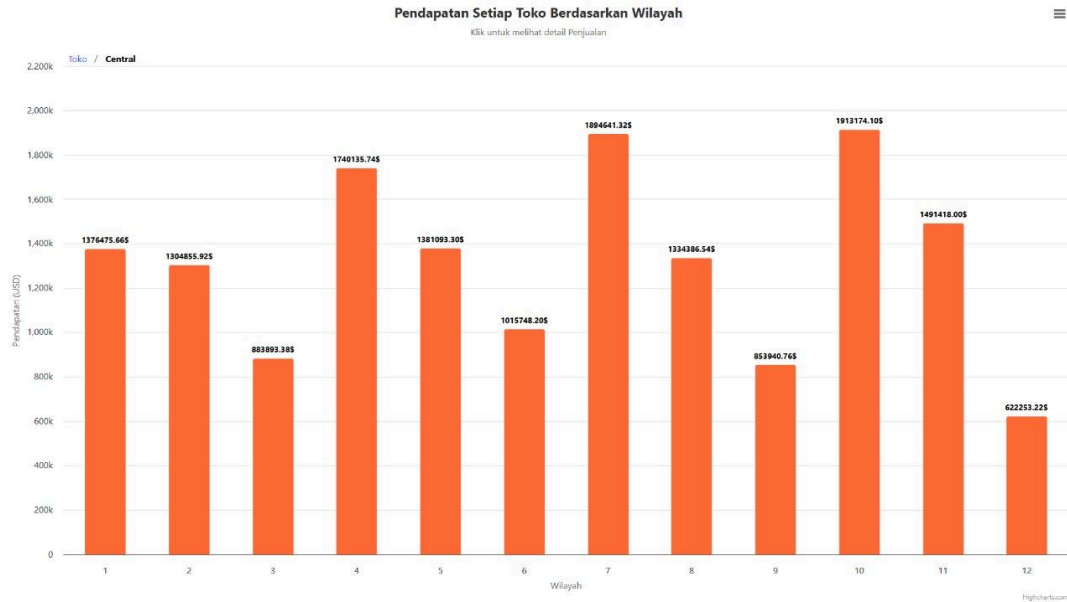


Grafik tersebut menampilkan pendapatan setiap toko berdasarkan wilayah dalam satuan USD. Wilayah dengan pendapatan tertinggi adalah **Southwest**, dengan total pendapatan mencapai **36,932,917.78 USD**, diikuti oleh **Canada** di posisi kedua dengan **28,755,851.34 USD**, dan **Northwest** di posisi ketiga dengan **24,870,152.00 USD**. Ketiga wilayah ini secara signifikan mendominasi pendapatan dibandingkan dengan wilayah lainnya.

Di kelompok menengah, wilayah **Central** dan **Southeast** memiliki pendapatan yang hampir sama, masing-masing sebesar **15,812,016.14 USD** dan **15,734,831.44 USD**. Wilayah **Northeast** berada sedikit di bawahnya dengan pendapatan **13,865,684.30 USD**.

Sementara itu, di kelompok dengan pendapatan terendah, terdapat **France** dengan pendapatan **9,215,075.74 USD**, diikuti oleh **United Kingdom** dengan **8,558,017.72 USD**. Dua wilayah dengan pendapatan paling rendah adalah **Germany** dengan **4,042,189.86 USD**, dan **Australia** di posisi terakhir dengan pendapatan sebesar **3,188,669.70 USD**.

Secara keseluruhan, grafik ini menunjukkan adanya kesenjangan yang cukup besar antara wilayah dengan pendapatan tertinggi dan terendah. Wilayah **Southwest** dan **Canada** menjadi kontributor utama, sementara **Australia** dan **Germany** memiliki kontribusi paling kecil terhadap total pendapatan.



Grafik tersebut menampilkan pendapatan setiap toko berdasarkan wilayah dalam satuan USD. Sumbu horizontal menunjukkan wilayah (dari 1 hingga 12), sementara sumbu vertikal menunjukkan jumlah pendapatan (dalam ribuan hingga jutaan USD). Bar oranye mewakili total pendapatan untuk setiap wilayah, dengan nilai numerik ditampilkan di atas masing-masing batang.

Secara umum, wilayah dengan pendapatan tertinggi adalah wilayah 10 (USD 1.913.174,10), diikuti oleh wilayah 7 (USD 1.894.631,32) dan wilayah 4 (USD 1.740.135,74). Sebaliknya, wilayah dengan pendapatan terendah adalah wilayah 12 (USD 622.253,22), diikuti oleh wilayah 9 (USD 853.940,76) dan wilayah 3 (USD 883.993,38). Grafik ini menunjukkan variasi pendapatan yang signifikan antar wilayah, yang dapat menjadi indikasi perbedaan performa atau potensi pasar di masing-masing wilayah.

## **BAB 4**

### **KESIMPULAN & SARAN**

#### **A. KESIMPULAN**

Proyek pengembangan data warehouse dan dashboard berbasis database Adventureworks 2019 berhasil menghasilkan sistem yang mampu mengintegrasikan data secara efektif untuk kebutuhan analisis penjualan dan pembelian. Proses ETL (Extract, Transform, Load) yang diterapkan memastikan data yang digunakan relevan, terstruktur, dan mudah dianalisis. Dashboard yang dibuat mempermudah manajemen dalam memahami pola penjualan, pembelian, serta efisiensi logistik. Analisis menunjukkan bahwa wilayah Amerika Utara mendominasi penjualan, kategori produk "Bikes" menjadi sumber utama pendapatan, dan metode pengiriman Cargo Transport memberikan kontribusi terbesar. Sistem ini membuktikan bahwa penggunaan teknologi data warehouse dan dashboard mampu meningkatkan kualitas pengambilan keputusan yang berbasis data

#### **B. SARAN**

Untuk meningkatkan efektivitas dan efisiensi, perusahaan disarankan untuk memperluas fokus ke wilayah dengan kontribusi rendah seperti Pasifik dan Eropa guna mengurangi ketergantungan pada wilayah utama. Inovasi dan promosi produk pada kategori yang kurang dominan, seperti Accessories dan Clothing, dapat membantu meningkatkan diversifikasi produk dan pendapatan. Selain itu, evaluasi metode pengiriman yang kurang optimal perlu dilakukan untuk meningkatkan efisiensi logistik. Perusahaan juga dapat mempertimbangkan pelatihan tambahan bagi karyawan untuk meningkatkan kinerja penjualan serta melakukan riset pasar guna memahami kebutuhan pelanggan yang lebih spesifik. Dengan strategi ini, perusahaan diharapkan dapat memperkuat daya saing dan mencapai pertumbuhan bisnis yang lebih baik

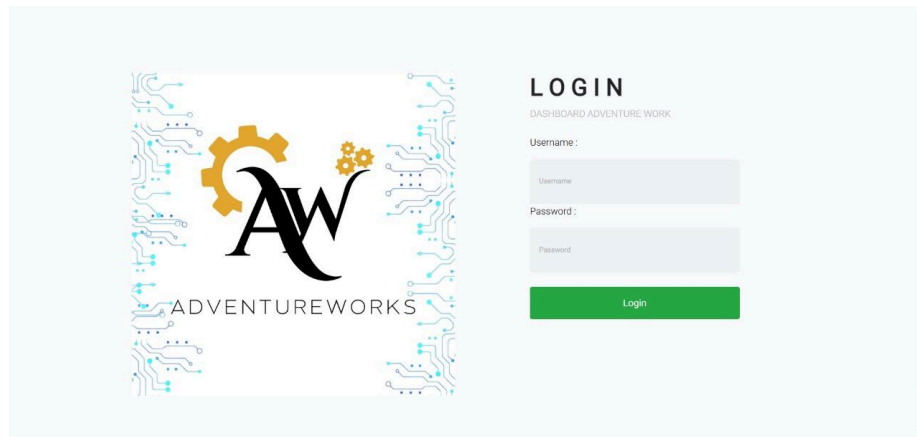


## Lampiran

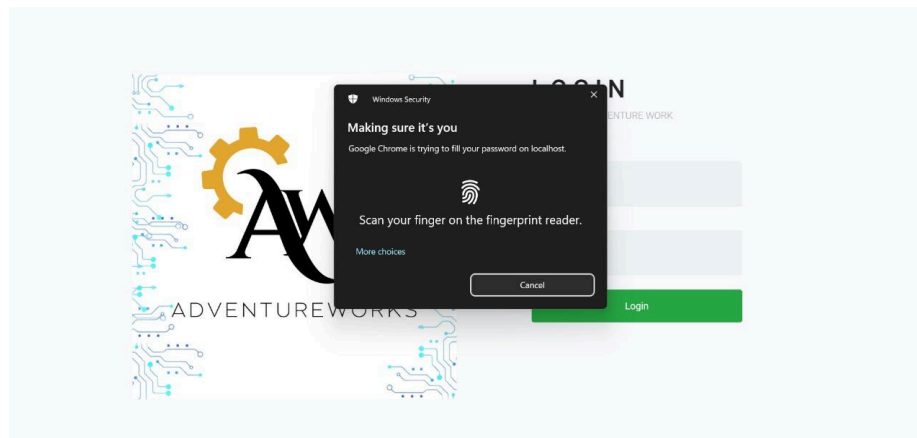
Link GitHub: // [https://github.com/kingzturch/FP\\_EAS\\_DWO.git](https://github.com/kingzturch/FP_EAS_DWO.git)

Dokumentasi Dashboard:

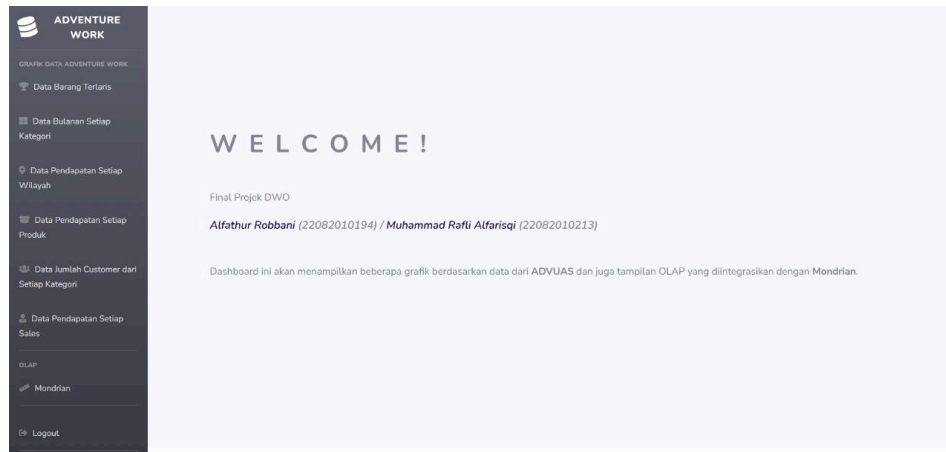
### Halaman Login



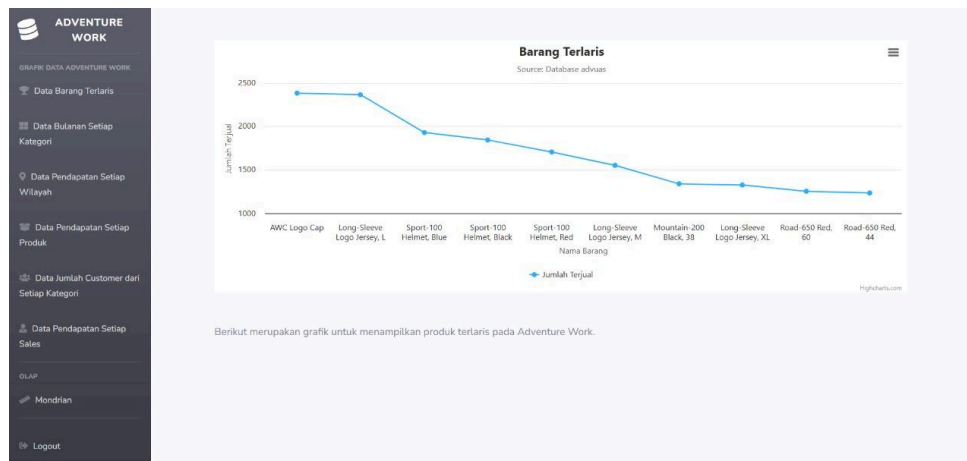
### Authentication



### Halaman Dashboard



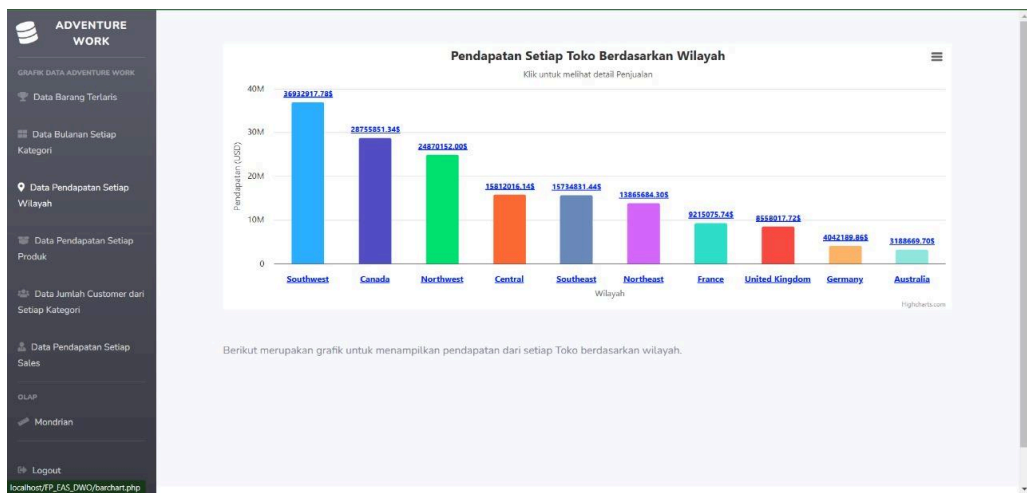
## Halaman Data Barang Terlaris



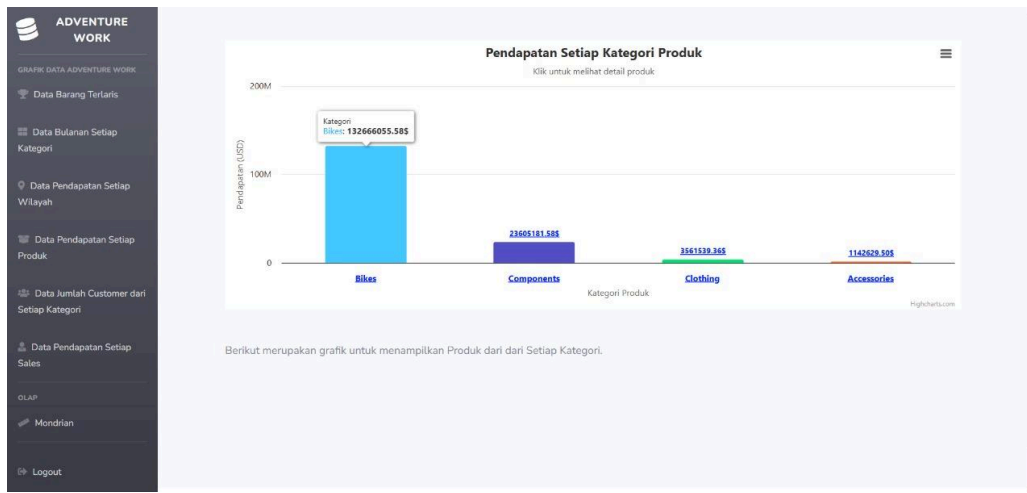
## Halaman Data Bulanan Setiap Kategori



## Halaman Data Pendapatan Setiap Toko Berdasarkan Wilayah



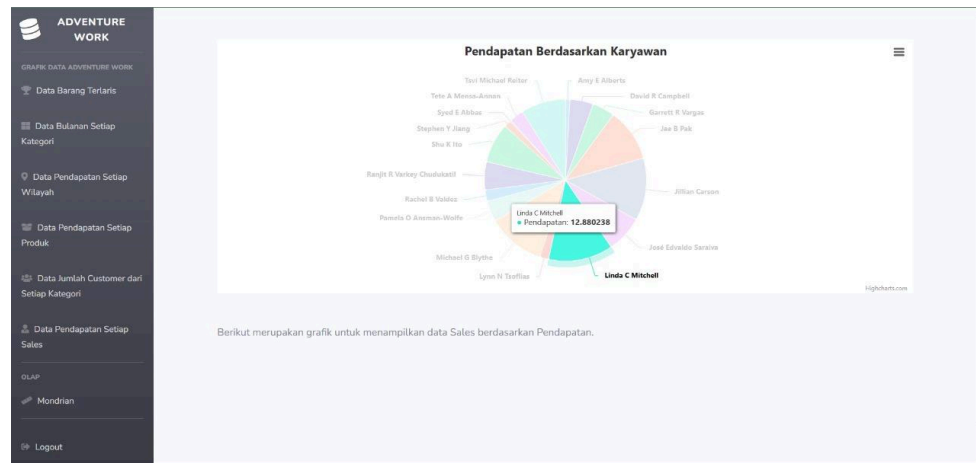
## Halaman Pendapatan Setiap Kategori Produk



## Halaman Data Jumlah Pelanggan Dari Setiap Kategori



## Halaman Pendapatan Berdasarkan Karyawan



## Halaman OLAP & Mondrian

