

PROJECT AKHIR
PENGEMBANGAN DATA WAREHOUSE DAN DASHBOARD
SEMESTER GANJIL 2024/2025



KELAS A

Roudhotul Izzah I **NPM : 22082010244**

Reihan Rachma Shafira **NPM : 22082010253**

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
Universitas Pembangunan Nasional “Veteran” Jawa Timur
S U R A B A Y A
2024

1. Deskripsi Studi Kasus

1.1 Penjelasan tentang Project “Pengembangan Data Warehouse dan Dashboard” yang dibangun.

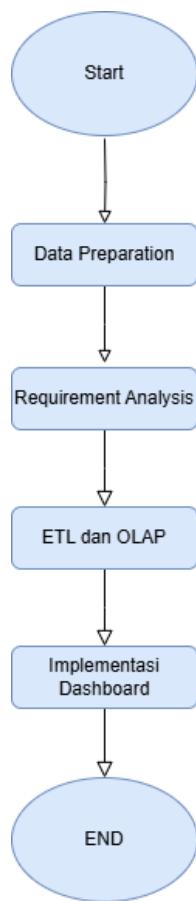
Kami mengembangkan sebuah proyek yang berfokus pada pembuatan tampilan dashboard data warehouse menggunakan basis data data warehouse bernama **projectuas_**. Data warehouse ini dibangun berdasarkan beberapa tabel yang diambil dari database **AdventureWorks2019**, termasuk tabel **Customer, Employee, Product, Sales Territory, Time, dan Scrap Reason**, yang digunakan sebagai dasar tabel dimensi. Dari tabel-tabel dimensi ini, kami menghasilkan tabel fakta seperti **Fakta Produk** dan **Fakta Penjualan** yang menyajikan data terukur untuk analisis.

Database **AdventureWorks2019** sendiri adalah contoh database bisnis yang dirancang untuk mensimulasikan operasi perusahaan manufaktur sepeda fiktif bernama AdventureWorks Cycles. Isinya mencakup berbagai aspek bisnis seperti data produk (sepeda, suku cadang, dan aksesoris), informasi pelanggan dan transaksi penjualan, manajemen karyawan, pemasok dan inventori, lokasi gudang, serta keuangan. Tujuan utama pengembangan data warehouse dari database ini adalah untuk menyajikan informasi yang lebih fokus, terstruktur, dan relevan sehingga mendukung analisis data secara efektif.

Dashboard yang kami kembangkan bersifat sederhana, dirancang dengan elemen-elemen dasar seperti **Bootstrap, CSS, dan JavaScript**, serta menggunakan font yang mudah dibaca untuk memudahkan pengguna. Karena fitur yang kami implementasikan cenderung terbatas, tampilan dashboard dibuat minimalis namun tetap fungsional, sehingga dapat menyajikan data secara jelas dan intuitif.

1.2 Metodologi

Proses pengembangan Data Warehouse dan Dashboard dalam studi kasus kami mencakup beberapa tahapan utama, yaitu Data Preparation, Requirement Analysis, ETL dan OLAP, serta Implementasi Dashboard. Seluruh proses tersebut diilustrasikan melalui diagram alur (flowchart) berikut:



Gambar 1.1 Alur Tahapan Pengerjaan

a. Data Preparation

Pada tahap ini, kami mempersiapkan data mentah dari sumber utama, yaitu database **AdventureWorks2019**, agar dapat digunakan untuk analisis lebih lanjut. Proses ini mencakup pengumpulan data dari sumber yang relevan, ekstraksi data menggunakan tools seperti DBeaver. Data yang telah dipersiapkan kemudian dirancang agar mudah diintegrasikan ke dalam Data Warehouse. Tahap ini juga melibatkan pengelompokan data sesuai dengan kebutuhan analisis, seperti tabel dimensi dan tabel fakta.

b. Requirement Analysis

Setelah data dipersiapkan, tahap berikutnya adalah melakukan analisis kebutuhan (requirement analysis). Proses ini bertujuan untuk memahami kebutuhan pengguna dan pemangku kepentingan terkait. Analisis ini mencakup identifikasi elemen-elemen penting yang perlu disajikan dalam dashboard, seperti jenis laporan, metrik utama (key metrics), dan kebutuhan visualisasi. Selain itu, tahap ini juga memastikan bahwa struktur data yang disiapkan sesuai dengan kebutuhan analisis pengguna, sehingga data warehouse yang dikembangkan dapat mendukung pengambilan keputusan secara efektif.

c. ETL dan OLAP

Tahap ini merupakan inti dari pengolahan data dalam pengembangan Data Warehouse, di mana proses **Extract, Transform, Load (ETL)** dilakukan untuk mentransfer data dari sumber aslinya ke dalam Data Warehouse. Pada tahap ekstraksi, data diambil dari berbagai tabel sumber, seperti tabel **Customer, Employee, Product, Sales Territory, Time, dan Scrap Reason**, menggunakan tools ETL seperti **Pentaho Data Integration (PDI)**. Setelah itu, data yang telah diambil diproses melalui transformasi, seperti normalisasi, denormalisasi, pengelompokan data, agregasi, perhitungan metrik, dan pembuatan hubungan antar tabel dimensi untuk menghasilkan tabel fakta. Tabel fakta ini berisi data terukur, seperti jumlah penjualan, total pendapatan, atau jumlah produk yang terjual, yang dihubungkan dengan tabel dimensi.

Setelah proses transformasi selesai, data dimuat (load) ke dalam Data Warehouse untuk mendukung analisis lebih lanjut. Proses ini dilanjutkan dengan penggunaan **OLAP (Online Analytical Processing)**, yang memungkinkan analisis data multidimensi secara fleksibel. Dengan OLAP, pengguna dapat melakukan eksplorasi data yang lebih mendalam untuk mendukung pengambilan keputusan strategis. Tools seperti Pentaho berperan penting dalam tahap ini karena mempermudah seluruh proses ETL hingga pembuatan tabel fakta secara sistematis dan efisien.

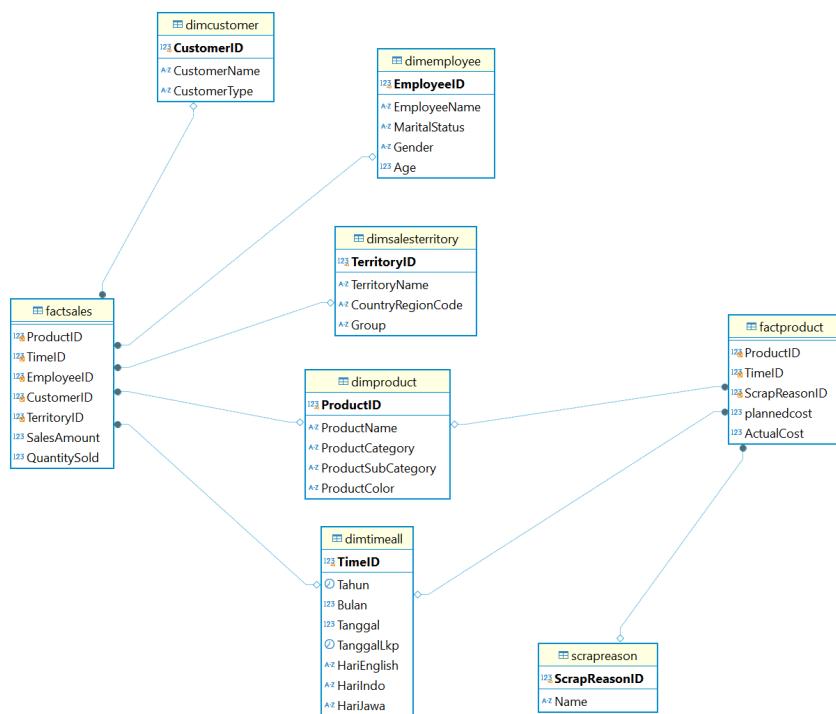
d. Implementasi Dashboard

Tahap ini berfokus pada pengembangan dan implementasi dashboard yang berfungsi untuk menyajikan data yang telah diproses dalam Data Warehouse. Dashboard dirancang menggunakan teknologi seperti Bootstrap, CSS, dan JavaScript untuk memastikan tampilan yang sederhana namun fungsional. Visualisasi data dibuat dengan mempertimbangkan kenyamanan pengguna, sehingga data dapat dipahami dengan mudah. Elemen-elemen utama yang ditampilkan dalam dashboard meliputi laporan berbasis grafik, tabel interaktif, dan metrik kunci yang relevan dengan kebutuhan pengguna.

2. Tahapan Project

Pada tahap ini, kami merancang data warehouse untuk menganalisis kinerja penjualan oleh pelanggan dan pembelian oleh karyawan. Desain data warehouse dapat dilihat pada Gambar 2.1 di bawah ini.

2.1 Implementasi Model Multidimensi



Gambar 2.1 Desain Star-Schema Data Warehouse Projectuas_

2.1.1 Tabel Dimensi

Berikut adalah susunan dari tabel dimensi :

1. Tabel dimcustomer

```
CREATE TABLE DimCustomer (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100),
    CustomerType CHAR(1)
);
```

2. Tabel dimemployee

```
CREATE TABLE DimEmployee (
    EmployeeID INT PRIMARY KEY,
    EmployeeName VARCHAR(100),
    MaritalStatus VARCHAR(1),
    Gender VARCHAR(1),
    Age INT
);
```

3. Tabel dimsalesterritory

```
CREATE TABLE DimSalesTerritory (
    TerritoryID INT PRIMARY KEY,
    TerritoryName VARCHAR(100),
    CountryRegionCode VARCHAR(5),
    `Group` VARCHAR(50) -- Menggunakan backtick
    untuk Group
);
```

4. Tabel dimproduct

```
CREATE TABLE DimProduct (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    ProductCategory VARCHAR(50),
    ProductSubCategory VARCHAR(50),
    ProductColor VARCHAR(30)
);
```

5. Tabel dimtimeall

```
CREATE TABLE DimTimeAll (
    TimeID INT PRIMARY KEY,
    Tahun YEAR,
    Bulan INT,
    Tanggal INT,
    TanggalLkp DATE,
    HariEnglish VARCHAR(15),
    HariIndo VARCHAR(15),
    HariJawa VARCHAR(15)
);
```

6. Tabel scrapreason

```
create table ScrapReason (
    ScrapReasonID INT PRIMARY key,
    Name VARCHAR(100)
);
```

2.1.2 Kebutuhan Data Tabel Dimensi dari AdventureWorks 2019.

1. Tabel dimcustomer

Data yang dimasukkan	Sumber data
CustomerID	Data diambil dari : Tabel sales_customer kolom PersonID → CustomerID
CustomerName	Data diambil dari : Tabel person_person kolom gabungan dari (FirstName + LastName) → CustomerName
CustomerType	Data diambil dari : Tabel person_person kolom PersonType → CustomerName

Query

```
INSERT INTO `projectuas_`.DimCustomer
(CustomerID, CustomerName, CustomerType)
SELECT
    sc.PersonID AS CustomerID,
    concat(pp.FirstName, " ", pp.LastName) as CustomerName,
    pp.PersonType as CustomerType
FROM
    sales_customer sc
INNER JOIN
    person_person pp ON sc.PersonID =
pp.BusinessEntityID
WHERE
    sc.PersonID IS NOT NULL;
```

2. Tabel dimemployee

Data yang dimasukkan	Sumber data
EmployeeID	Data diambil dari : Tabel sales_salesperson kolom BusinessEntityID → EmployeeID
EmployeeName	Data diambil dari :

	Tabel person_person kolom gabungan dari (FirstName + LastName) → EmployeeName
MaritalStatus	Data diambil dari : Tabel humanresources_employee kolom MaritalStatus → MaritalStatus
Gender	Data diambil dari : Tabel humanresources_employee kolom Gender → Gender
Age	Data diambil dari : Tabel humanresources_employee kolom BirthDate (acuan '2001-01-01') → Age

Query

```

INSERT INTO `projectuas_`.DimEmployee
(EmployeeID, EmployeeName, MaritalStatus,
Gender, Age)
SELECT
    ss.BusinessEntityID as EmployeeID ,
    concat(pp.FirstName, " ", pp.LastName) as EmployeeName,
    he.MaritalStatus as MaritalStatus,
    he.Gender as Gender,
    TIMESTAMPDIFF(YEAR, he.BirthDate,
    '2001-01-01') AS Age
FROM
    sales_salesperson ss
join
humanresources_employee he on
ss.BusinessEntityID = he.BusinessEntityID
join
person_person pp on pp.BusinessEntityID =
he.BusinessEntityID

```

3. Tabel dimsaleterritory

Data yang dimasukkan	Sumber data
TerritoryID	Data diambil dari :

	Tabel sales_territory kolom TerritoryID → TerritoryID
TerritoryName	Data diambil dari : Tabel sales_territory kolom Name → TerritoryName
CountryRegionCode	Data diambil dari : Tabel sales_territory kolom CountryRegionCode → CountryRegionCode
Group	Data diambil dari : Tabel sales_territory kolom Group → Group

Query

```

INSERT INTO `projectuas_`.dimsalesterritory
(TerritoryID, TerritoryName,
CountryRegionCode, `Group`)
SELECT
    st.TerritoryID,
    st.Name AS TerritoryName,
    st.CountryRegionCode,
    st.`Group` AS GroupAlias
FROM
    sales_salesterritory st;

```

4. Tabel dimproduct

Data yang dimasukkan	Sumber data
ProductID	Data diambil dari : Tabel production_product kolom ProductID → ProductID
ProductName	Data diambil dari : Tabel production_product kolom Name → ProductName
ProductCategory	Data diambil dari : Tabel production_productcategory kolom Name → ProductCategory

ProductSubCategory	Data diambil dari : Tabel production_productsubcategory kolumn Name → ProductSubCategory
ProductColor	Data diambil dari : Tabel production_product kolumn Color → ProductColor

Query

```


INSERT INTO `Projectuas_`.DimProduct
(ProductID, ProductName, ProductCategory,
ProductSubCategory, ProductColor)
SELECT
    p.ProductID,
    p.Name AS ProductName,
    pc.Name AS ProductCategory,
    psc.Name AS ProductSubCategory,
    p.Color AS ProductColor
FROM
    production_product p
JOIN
    production_productsubcategory psc ON
p.ProductSubcategoryID =
psc.ProductSubcategoryID
JOIN
    production_productcategory pc ON
psc.ProductCategoryID = pc.ProductCategoryID;


```

5. Tabel dimtimeall

Tabel **dimtime** hanya membutuhkan data waktu hingga tingkat harian (tanggal). Sebagai tabel dimensi, dimtime berfungsi sebagai acuan dan tidak bergantung pada data transaksi. Namun, baris pertama tabel ini perlu merujuk pada data transaksi di tabel **SalesOrderDetail**, khususnya kolom **ModifiedDate**, dalam database AdventureWorks.

Data dalam tabel **dimtime** akan mencakup rentang waktu dari 16 April 2014 hingga 22 September 2014. Setelah rentang waktu ditentukan,

data diisi menggunakan sebuah Stored Procedure bernama **isitime**.

Query untuk mengisi data tersebut dimasukkan ke dalam procedure ini:

```
CREATE DEFINER='root'@'localhost' PROCEDURE
`projectuas_`.`isitimeall`(IN `waktuMulai` 
DATE, IN `waktuSelesai` DATE, IN `HariJawa` 
VARCHAR(20), IN `HariIndo` VARCHAR(20))
begin

    -- Label untuk loop
    mulaiLoop: LOOP
        -- Memasukkan data ke dalam tabel
        dimtime
            INSERT INTO dimtimeall (Tahun, Bulan,
Tanggal, TanggalLkp, HariEnglish, HariIndo,
HariJawa)
            VALUES (
                YEAR(waktuMulai),
                MONTH(waktuMulai),
                DAY(waktuMulai),
                waktuMulai,
                DAYNAME(waktuMulai),
                HariIndo,
                HariJawa
            );
        -- Menambah satu hari ke waktuMulai
        SET waktuMulai = DATE_ADD(waktuMulai,
INTERVAL 1 DAY);
        -- Mengubah HariJawa berdasarkan siklus
        5 hari
        SET HariJawa =
            CASE
                WHEN HariJawa = 'pon' THEN
                'wage'
                WHEN HariJawa = 'wage' THEN
                'kliwon'
                WHEN HariJawa = 'kliwon' THEN
                'legi'
                WHEN HariJawa = 'legi' THEN
                'pahing'
                WHEN HariJawa = 'pahing' THEN
                'pon'
```

```

        END;
        -- Mengubah HariIndo berdasarkan siklus
4 hari
        SET HariIndo =
        CASE
            WHEN HariIndo = 'Minggu' THEN
            'Senin'
            WHEN HariIndo = 'Senin' THEN
            'Selasa'
            WHEN HariIndo = 'Selasa' THEN
            'Rabu'
            WHEN HariIndo = 'Rabu' THEN
            'Kamis'
            WHEN HariIndo = 'Kamis' THEN
            'Jumat'
            WHEN HariIndo = 'Jumat' THEN
            'Sabtu'
            WHEN HariIndo = 'Sabtu' THEN
            'Minggu'
        END;
        -- Memeriksa apakah waktuMulai melebihi
waktuSelesai
        IF DATEDIFF(waktuMulai, waktuSelesai) >
0 THEN
            LEAVE mulaiLoop; -- Keluar dari
loop jika waktuMulai sudah lebih dari
waktuSelesai
        END IF;

    END LOOP mulaiLoop;
END

```

Setelah Stored Procedure `isitime` dibuat dan disimpan, langkah selanjutnya adalah menjalankannya untuk mengisi data dengan memanggil prosedur tersebut.

```

CALL      projectuas_.isitimeall      ('2011-04-16',
'2014-09-22', 'wage', 'sabtu');

```

6. Tabel scrapreason

Data yang dimasukkan	Sumber data
ScrapReasonID	Data diambil dari : Tabel prouction_scrapreason kolom ScrapReasonID→ ScrapReasonID
Name	Data diambil dari : Tabel prouction_scrapreason kolom Name → Name

Query

```
INSERT INTO
`projectuas_`.scrapreason(ScrapReasonID, Name)
select
    ps.ScrapReasonID ,
    ps.Name
from
    production_scrapreason ps;
```

2.1.3 Tabel Fakta

1. Tabel factsales

```
CREATE TABLE FactSales (
    ProductID INT,
    TimeID INT,
    EmployeeID INT,
    CustomerID INT,
    TerritoryID INT,
    SalesAmount DOUBLE,
    QuantitySold SMALLINT(6),
    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)
);

```

2. Tabel factproduct

```

CREATE TABLE FactProduct (
    ProductID INT,
    TimeID INT,
    scrapreasonID INT,
    plannedcost Double,
    ActualCost Double,
    FOREIGN KEY (ProductID) REFERENCES
DimProduct(ProductID),
    FOREIGN KEY (TimeID) REFERENCES
DimTime(TimeID),
    FOREIGN KEY (scrapreasonID) REFERENCES
ScrapReason(scrapreasonID)
);

```

3. Hubungan Antara Tabel Fakta Dan Tabel Dimensi

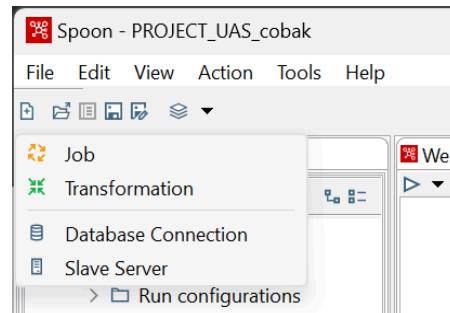
- “Seorang karyawan bernama **Stephen Y Jiang** berhasil menjual **2** unit **Mountain-100 Black, 42** kepada pelanggan bernama **Roger Harui** pada tanggal **26 Oktober 2024**. Transaksi tersebut dilakukan di wilayah **Northwest, US** dengan total penjualan mencapai **\$ 4049.988**.”
- “Pada tanggal **4 Juli 2011**, produk **LL Road Seat Assembly** dikembalikan ke perusahaan karena alasan **perakitan rem**

yang tidak sesuai pesanan. Akibatnya, perusahaan mengalami kerugian sebesar **\$36.75** dari perencanaan penjualan produk tersebut, dan produk yang dikembalikan tersebut direalisasikan dengan harga **\$36.75.**"

2.2 Implementasi Cube dan OLAP

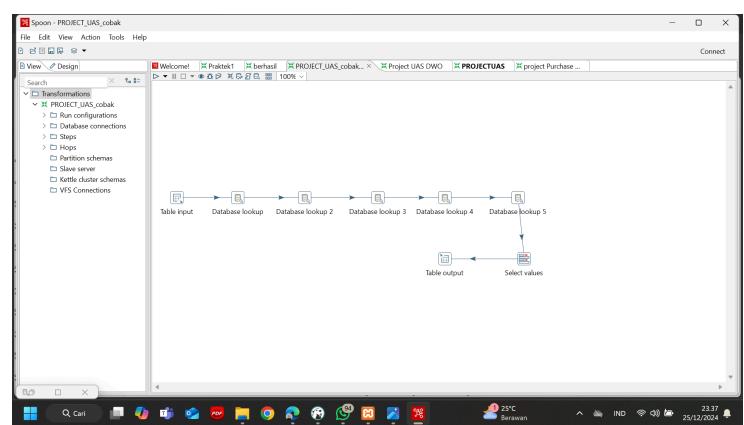
Pada tahap ini, dilakukan integrasi data menggunakan tools Pentaho. Berikut adalah langkah - langkahnya :

- 1) Alur implementasi ETL & Olap untuk tabel factsales
 - a. Mengkoneksikan database
 - Mulailah dengan membuat file transformasi baru dan beri nama sesuai kebutuhan.



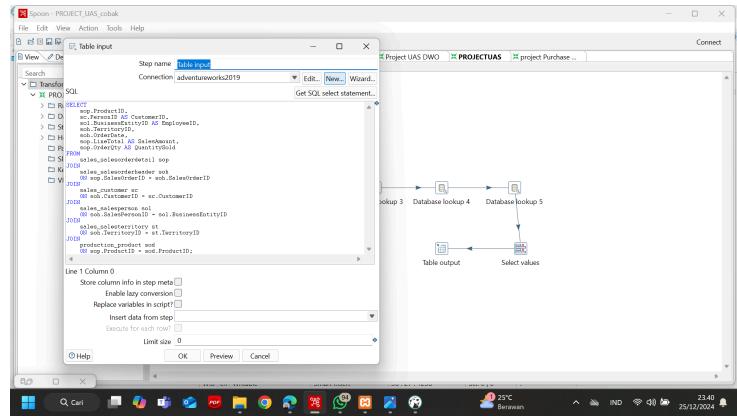
Gambar 2.1 Icon Buat Transformasi

- Desain transformasi sesuaikan dengan gambar di bawah ini :



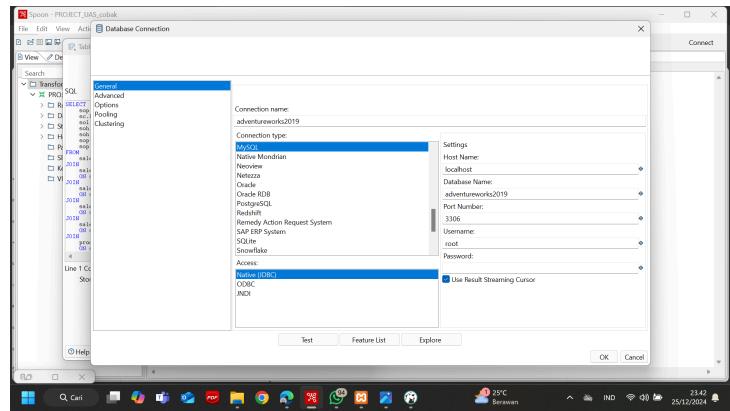
Gambar 2.2 Desain Transformasi factsales

- Klik dua kali pada tabel input, kemudian pilih "New."



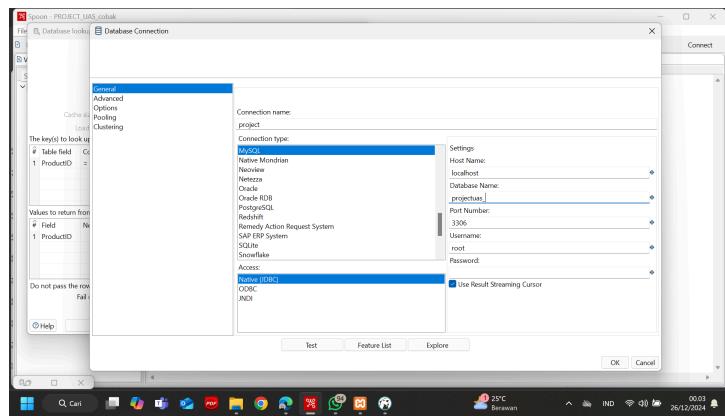
Gambar 2.3 Tambah Database

- Tentukan koneksi dengan database AdventureWorks, seperti yang ditunjukkan pada gambar 2.2.4 dibawah ini. Lalu klik Test dan jika berhasil klik OK.



Gambar 2.4 Koneksi Database Adventureworks

- Lalu lakukan hal yang sama pada lookup 1,2,3,4, dan 5 tapi hubungkan dengan database projectuas_ yang telah dibuat sebelumnya.



Gambar 2.5 Koneksi Database Projectus_

b. Query tabel input

Pada tabel input masukkan query seperti dibawah ini :

```
SELECT
    sop.ProductID,
    sc.PersonID AS CustomerID,
    sol.BusinessEntityID AS EmployeeID,
    soh.TerritoryID,
    soh.OrderDate,
    sop.LineTotal AS SalesAmount,
    sop.OrderQty AS QuantitySold
FROM
    sales_salesorderdetail sop
JOIN
    sales_salesorderheader soh
    ON sop.SalesOrderID =
    soh.SalesOrderID
JOIN
    sales_customer sc
```

```

    ON soh.CustomerID = sc.CustomerID

JOIN

    sales_salesperson sol

    ON soh.SalesPersonID =
sol.BusinessEntityID

JOIN

    sales_salesterritory st

    ON soh.TerritoryID = st.TerritoryID

JOIN

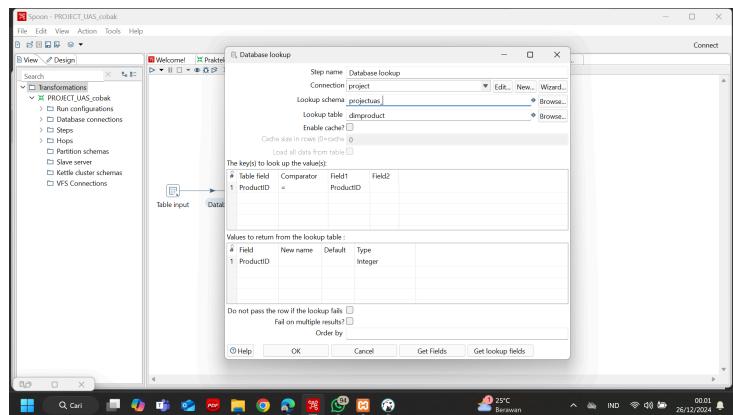
    production_product sod

    ON sop.ProductID = sod.ProductID;

```

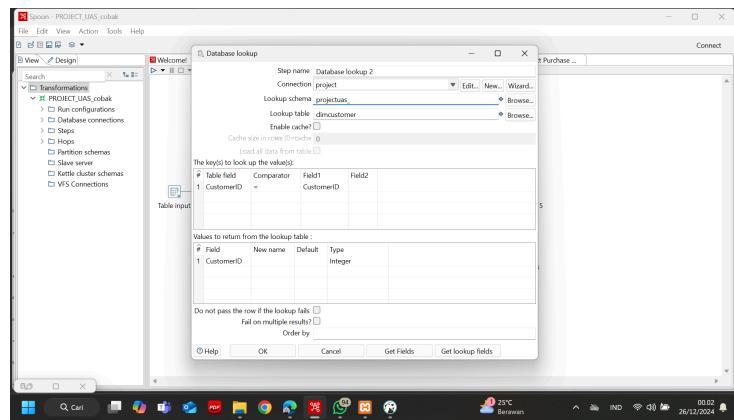
c. Tahapan lookup dan lainnya

- Pada database lookup 1 memasukkan data dari ProductID, sesuaikan dengan gambar dibawah ini :



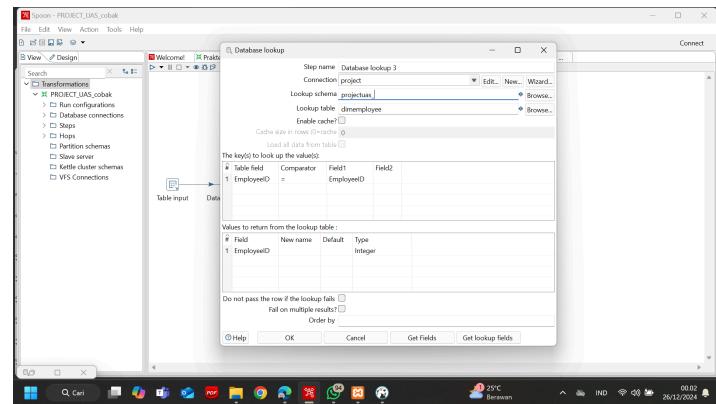
Gambar 2.6 input database lookup 1 factsales

- Pada database lookup 2 memasukkan data dari CustomerID, sesuaikan dengan gambar dibawah ini :



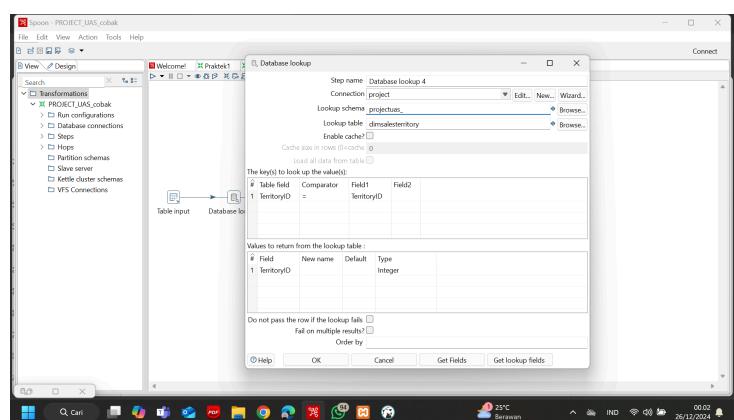
Gambar 2.7 Input database lookup 2 factsales

- Pada database lookup 3 memasukkan data dari EmployeeID, sesuaikan dengan gambar dibawah ini :



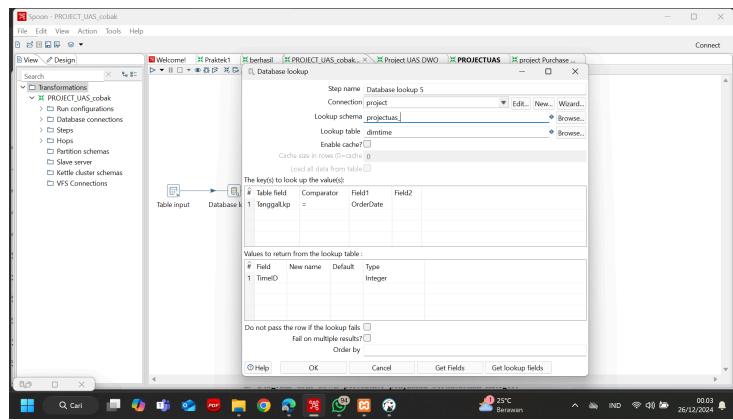
Gambar 2.8 Input database lookup 3 factsales

- Pada database lookup 4 memasukkan data dari TerritoryID, sesuaikan dengan gambar dibawah ini :



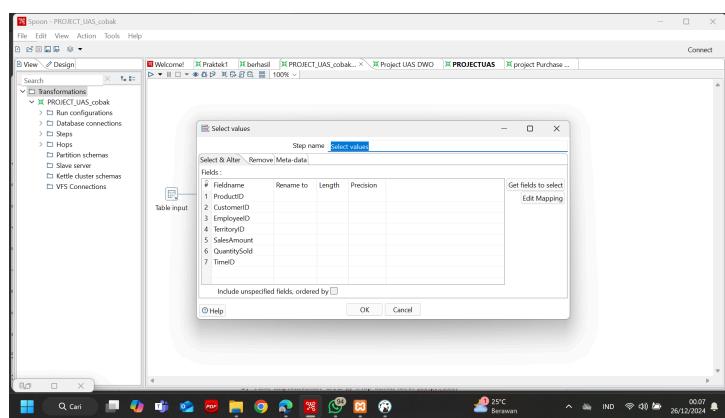
Gambar 2.9 Input database lookup 4 factsales

- Pada database lookup 5 memasukkan data dari TimetID, sesuaikan dengan gambar dibawah ini :



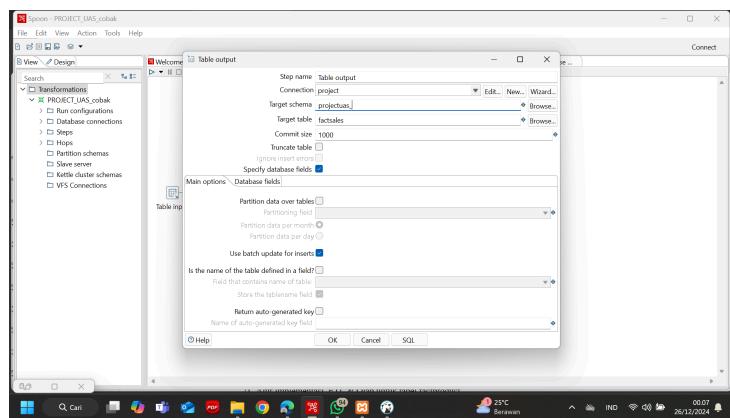
Gambar 2.10 Input database lookup 5 factsales

- Selanjutnya, pada bagian select value, sesuaikan seperti pada gambar di bawah ini :

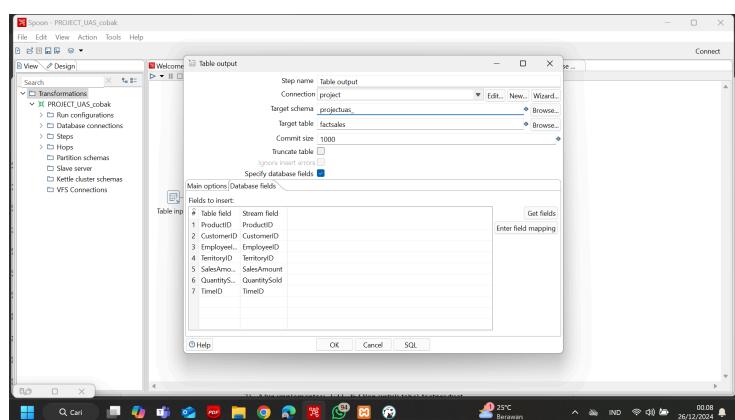


Gambar 2.11 Input select value factsales

- Lalu yang terakhir, pada tabel output sesuaikan dengan gambar di bawah ini :



**Gambar 2.11 Input tabel output main options
factsales**



**Gambar 2.12 Input tabel output Database fields
factsales**

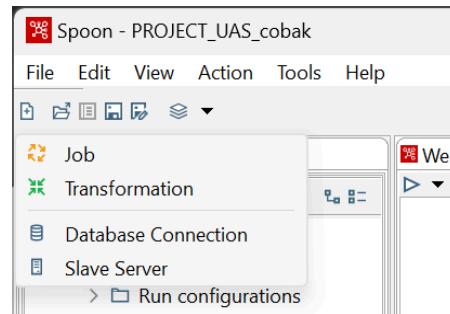
- Setelah itu lakukan run lalu tunggu. Jika berhasil, data akan otomatis masuk ke database dan dapat dilihat di Dbeaver.

factsales	ProductID	CategoryID	Description	UnitPrice	UnitsInStock	UnitPrice
1	1	1	Chai	18.0	39	18.0
2	1	1	Chai	18.0	39	18.0
3	1	1	Chai	18.0	39	18.0
4	1	1	Chai	18.0	39	18.0
5	1	1	Chai	18.0	39	18.0
6	1	1	Chai	18.0	39	18.0
7	1	1	Chai	18.0	39	18.0
8	1	1	Chai	18.0	39	18.0
9	1	1	Chai	18.0	39	18.0
10	1	1	Chai	18.0	39	18.0
11	1	1	Chai	18.0	39	18.0
12	1	1	Chai	18.0	39	18.0
13	1	1	Chai	18.0	39	18.0
14	1	1	Chai	18.0	39	18.0
15	1	1	Chai	18.0	39	18.0
16	1	1	Chai	18.0	39	18.0
17	1	1	Chai	18.0	39	18.0
18	1	1	Chai	18.0	39	18.0
19	1	1	Chai	18.0	39	18.0
20	1	1	Chai	18.0	39	18.0
21	1	1	Chai	18.0	39	18.0
22	1	1	Chai	18.0	39	18.0
23	1	1	Chai	18.0	39	18.0
24	1	1	Chai	18.0	39	18.0
25	1	1	Chai	18.0	39	18.0
26	1	1	Chai	18.0	39	18.0
27	1	1	Chai	18.0	39	18.0
28	1	1	Chai	18.0	39	18.0
29	1	1	Chai	18.0	39	18.0
30	1	1	Chai	18.0	39	18.0
31	1	1	Chai	18.0	39	18.0
32	1	1	Chai	18.0	39	18.0
33	1	1	Chai	18.0	39	18.0
34	1	1	Chai	18.0	39	18.0
35	1	1	Chai	18.0	39	18.0
36	1	1	Chai	18.0	39	18.0
37	1	1	Chai	18.0	39	18.0
38	1	1	Chai	18.0	39	18.0
39	1	1	Chai	18.0	39	18.0
40	1	1	Chai	18.0	39	18.0
41	1	1	Chai	18.0	39	18.0
42	1	1	Chai	18.0	39	18.0
43	1	1	Chai	18.0	39	18.0
44	1	1	Chai	18.0	39	18.0
45	1	1	Chai	18.0	39	18.0
46	1	1	Chai	18.0	39	18.0
47	1	1	Chai	18.0	39	18.0
48	1	1	Chai	18.0	39	18.0
49	1	1	Chai	18.0	39	18.0
50	1	1	Chai	18.0	39	18.0
51	1	1	Chai	18.0	39	18.0
52	1	1	Chai	18.0	39	18.0
53	1	1	Chai	18.0	39	18.0
54	1	1	Chai	18.0	39	18.0
55	1	1	Chai	18.0	39	18.0
56	1	1	Chai	18.0	39	18.0
57	1	1	Chai	18.0	39	18.0
58	1	1	Chai	18.0	39	18.0
59	1	1	Chai	18.0	39	18.0
60	1	1	Chai	18.0	39	18.0
61	1	1	Chai	18.0	39	18.0
62	1	1	Chai	18.0	39	18.0
63	1	1	Chai	18.0	39	18.0
64	1	1	Chai	18.0	39	18.0
65	1	1	Chai	18.0	39	18.0
66	1	1	Chai	18.0	39	18.0
67	1	1	Chai	18.0	39	18.0
68	1	1	Chai	18.0	39	18.0
69	1	1	Chai	18.0	39	18.0
70	1	1	Chai	18.0	39	18.0
71	1	1	Chai	18.0	39	18.0
72	1	1	Chai	18.0	39	18.0
73	1	1	Chai	18.0	39	18.0
74	1	1	Chai	18.0	39	18.0
75	1	1	Chai	18.0	39	18.0
76	1	1	Chai	18.0	39	18.0
77	1	1	Chai	18.0	39	18.0
78	1	1	Chai	18.0	39	18.0
79	1	1	Chai	18.0	39	18.0
80	1	1	Chai	18.0	39	18.0
81	1	1	Chai	18.0	39	18.0
82	1	1	Chai	18.0	39	18.0
83	1	1	Chai	18.0	39	18.0
84	1	1	Chai	18.0	39	18.0
85	1	1	Chai	18.0	39	18.0
86	1	1	Chai	18.0	39	18.0
87	1	1	Chai	18.0	39	18.0
88	1	1	Chai	18.0	39	18.0
89	1	1	Chai	18.0	39	18.0
90	1	1	Chai	18.0	39	18.0
91	1	1	Chai	18.0	39	18.0
92	1	1	Chai	18.0	39	18.0
93	1	1	Chai	18.0	39	18.0
94	1	1	Chai	18.0	39	18.0
95	1	1	Chai	18.0	39	18.0
96	1	1	Chai	18.0	39	18.0
97	1	1	Chai	18.0	39	18.0
98	1	1	Chai	18.0	39	18.0
99	1	1	Chai	18.0	39	18.0
100	1	1	Chai	18.0	39	18.0
101	1	1	Chai	18.0	39	18.0
102	1	1	Chai	18.0	39	18.0
103	1	1	Chai	18.0	39	18.0
104	1	1	Chai	18.0	39	18.0
105	1	1	Chai	18.0	39	18.0
106	1	1	Chai	18.0	39	18.0
107	1	1	Chai	18.0	39	18.0
108	1	1	Chai	18.0	39	18.0
109	1	1	Chai	18.0	39	18.0
110	1	1	Chai	18.0	39	18.0
111	1	1	Chai	18.0	39	18.0
112	1	1	Chai	18.0	39	18.0
113	1	1	Chai	18.0	39	18.0
114	1	1	Chai	18.0	39	18.0
115	1	1	Chai	18.0	39	18.0
116	1	1	Chai	18.0	39	18.0
117	1	1	Chai	18.0	39	18.0
118	1	1	Chai	18.0	39	18.0
119	1	1	Chai	18.0	39	18.0
120	1	1	Chai	18.0	39	18.0
121	1	1	Chai	18.0	39	18.0
122	1	1	Chai	18.0	39	18.0
123	1	1	Chai	18.0	39	18.0
124	1	1	Chai	18.0	39	18.0
125	1	1	Chai	18.0	39	18.0
126	1	1	Chai	18.0	39	18.0
127	1	1	Chai	18.0	39	18.0
128	1	1	Chai	18.0	39	18.0
129	1	1	Chai	18.0	39	18.0
130	1	1	Chai	18.0	39	18.0
131	1	1	Chai	18.0	39	18.0
132	1	1	Chai	18.0	39	18.0
133	1	1	Chai	18.0	39	18.0
134	1	1	Chai	18.0	39	18.0
135	1	1	Chai	18.0	39	18.0
136	1	1	Chai	18.0	39	18.0
137	1	1	Chai	18.0	39	18.0
138	1	1	Chai	18.0	39	18.0
139	1	1	Chai	18.0	39	18.0
140	1	1	Chai	18.0	39	18.0
141	1	1	Chai	18.0	39	18.0
142	1	1	Chai	18.0	39	18.0
143	1	1	Chai	18.0	39	18.0
144	1	1	Chai	18.0	39	18.0
145	1	1	Chai	18.0	39	18.0
146	1	1	Chai	18.0	39	18.0
147	1	1	Chai	18.0	39	18.0
148	1	1	Chai	18.0	39	18.0
149	1	1	Chai	18.0	39	18.0
150	1	1	Chai	18.0	39	18.0
151	1	1	Chai	18.0	39	18.0
152	1	1	Chai	18.0	39	18.0
153	1	1	Chai	18.0	39	18.0
154	1	1	Chai	18.0	39	18.0
155	1	1	Chai	18.0	39	18.0
156	1	1	Chai	18.0	39	18.0
157	1	1	Chai	18.0	39	18.0
158	1	1	Chai	18.0	39	18.0
159	1	1	Chai	18.0	39	18.0
160	1	1	Chai	18.0	39	18.0
161	1	1	Chai	18.0	39	18.0
162	1	1	Chai	18.0	39	18.0
163	1	1	Chai	18.0	39	18.0
164	1	1	Chai	18.0	39	18.0
165	1	1	Chai	18.0	39	18.0
166	1	1	Chai	18.0	39	18.0
167	1	1	Chai	18.0	39	18.0
168	1	1	Chai	18.0	39	18.0
169	1	1	Chai	18.0	39	18.0
170	1	1	Chai	18.0	39	18.0
171	1	1	Chai	18.0	39	18.0
172	1	1	Chai	18.0	39	18.0
173	1	1	Chai	18.0	39	18.0
174	1	1	Chai	18.0	39	18.0
175	1	1	Chai	18.0	39	18.0
176	1	1	Chai	18.0	39	18.0
177	1	1	Chai	18.0	39	18.0
178	1	1	Chai	18.0	39	18.0
179	1	1	Chai	18.0	39	18.0
180	1	1	Chai	18.0	39	18.0
181	1	1	Chai	18.0	39	18.0
182	1	1	Chai	18.0	39	18.0
183	1	1	Chai	18.0	39	18.0
184	1	1	Chai	18.0	39	18.0
185	1	1	Chai	18.0	39	18.0
186	1	1	Chai	18.0	39	18.0
187	1	1	Chai	18.0	39	18.0
188	1	1</td				

2) Alur implementasi ETL & Olap untuk tabel factproduct

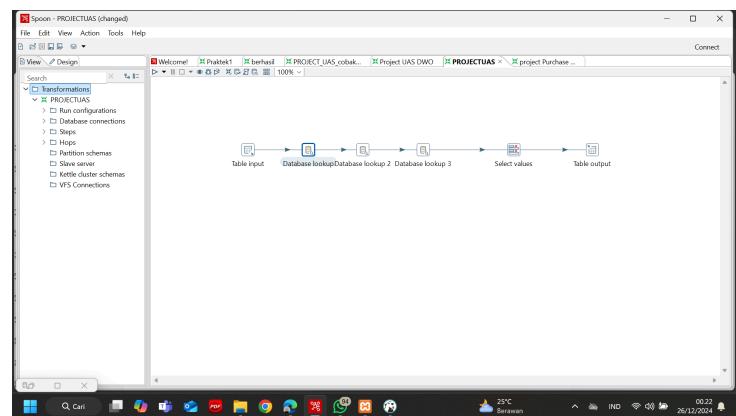
a. Mengkoneksikan database

- Mulailah dengan membuat file transformasi baru dan beri nama sesuai kebutuhan.



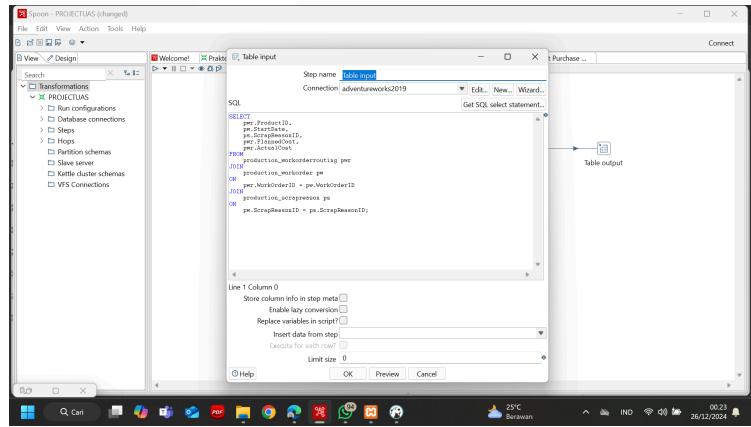
Gambar 2.14 Icon Buat Transformasi

- Desain transformasi sesuaikan dengan gambar di bawah ini :



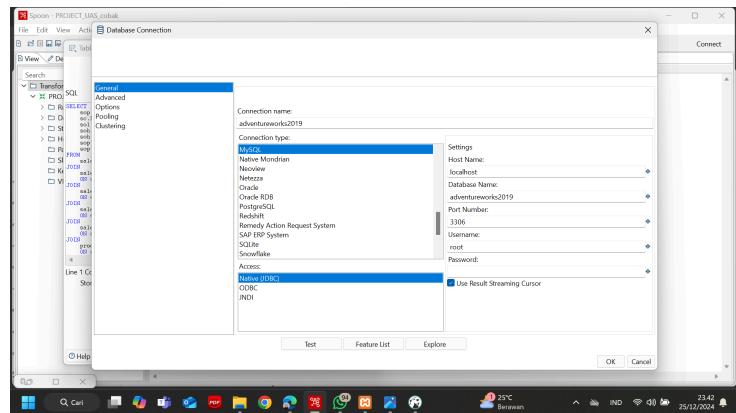
Gambar 2.15 Desain Transformasi factproduct

- Klik dua kali pada tabel input, kemudian pilih "New."



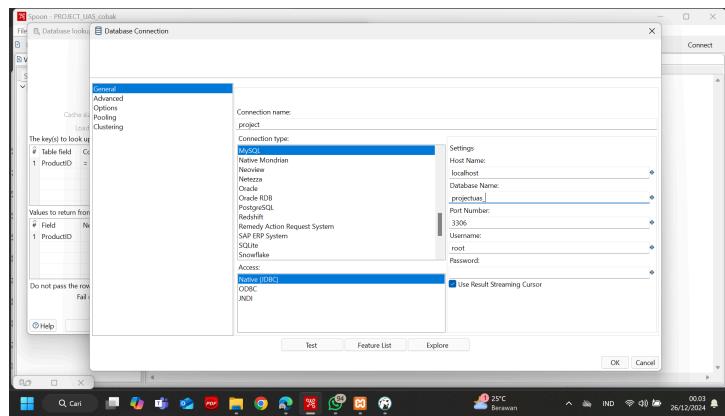
Gambar 2.16 Tambah Database

- Tentukan koneksi dengan database AdventureWorks, seperti yang ditunjukkan pada gambar dibawah ini. Lalu klik Test dan jika berhasil klik OK.



Gambar 2.17 Koneksi Database Adventureworks

- Lalu lakukan hal yang sama pada lookup 1,2, dan 3 tapi hubungkan dengan database projectuas_ yang telah dibuat sebelumnya.



Gambar 2.18 Koneksi Database Projectuas_

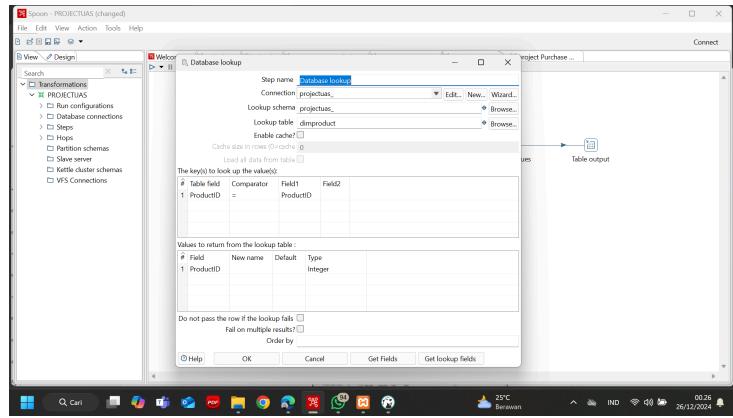
d. Query tabel input

Pada tabel input masukkan query seperti dibawah ini :

```
SELECT
    pwr.ProductID,
    pw.StartDate,
    ps.ScrapReasonID,
    pwr.PlannedCost,
    pwr.ActualCost
FROM
    production_workorderrouting pwr
JOIN
    production_workorder pw
ON
    pwr.WorkOrderID = pw.WorkOrderID
JOIN
    production_scrapreason ps
ON
    pw.ScrapReasonID = ps.ScrapReasonID;
```

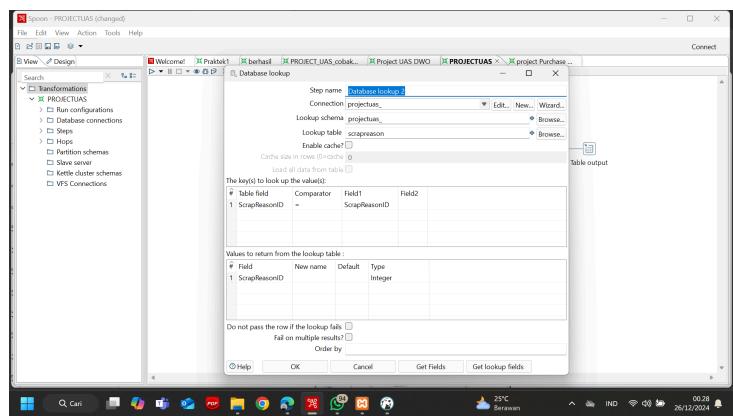
e. Tahapan lookup dan lainnya

- Pada database lookup 1 memasukkan data dari ProductID, sesuaikan dengan gambar dibawah ini :



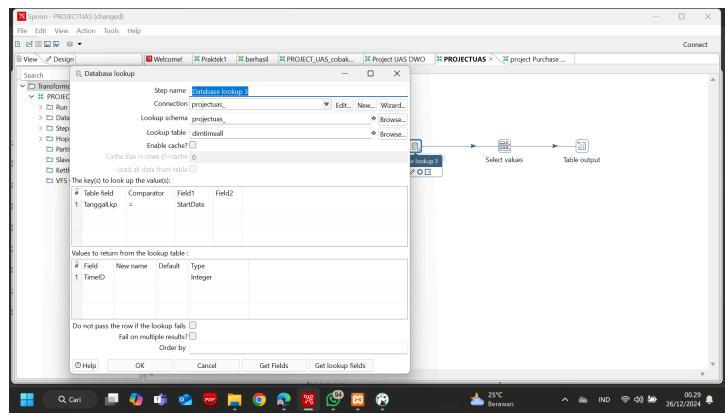
Gambar 2.19 input database lookup 1 factproduct

- Pada database lookup 2 memasukkan data dari ScrapReasonID, sesuaikan dengan gambar dibawah ini :



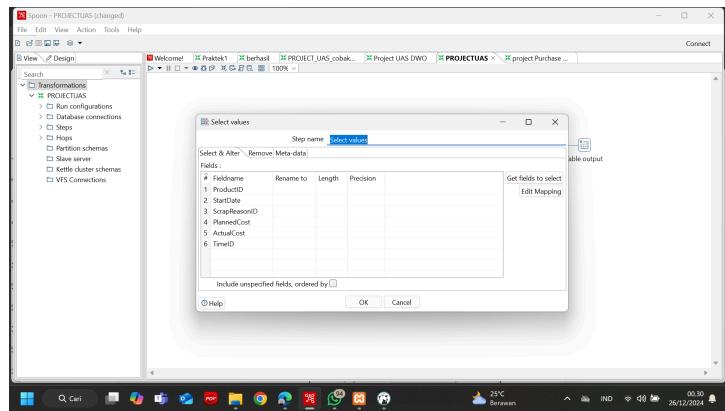
Gambar 2.20 Input database lookup 2 factproduct

- Pada database lookup 3 memasukkan data dari TimeID, sesuaikan dengan gambar dibawah ini :



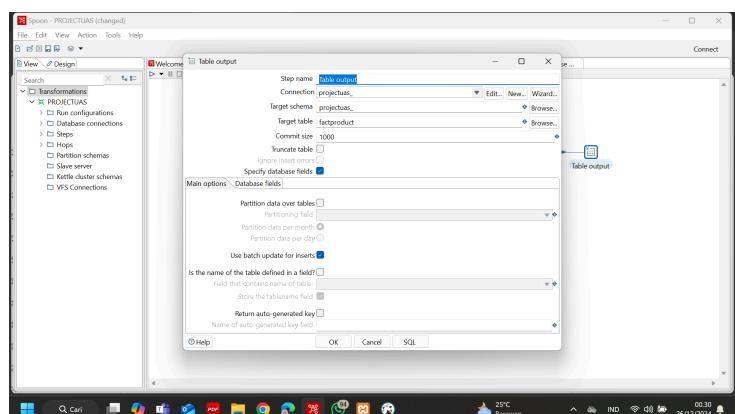
Gambar 2.21 Input database lookup 3 factproduct

- Selanjutnya, pada bagian select value, sesuaikan seperti pada gambar di bawah ini :

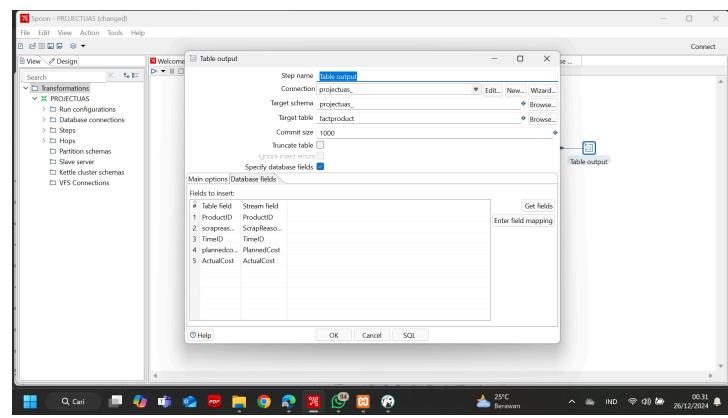


Gambar 2.22 Input select value factproduct

- Lalu yang terakhir, pada tabel output sesuaikan dengan gambar di bawah ini :



**Gambar 2.23 Input tabel output main options
factproduct**



**Gambar 2.24 Input tabel output Database fields
factproduct**

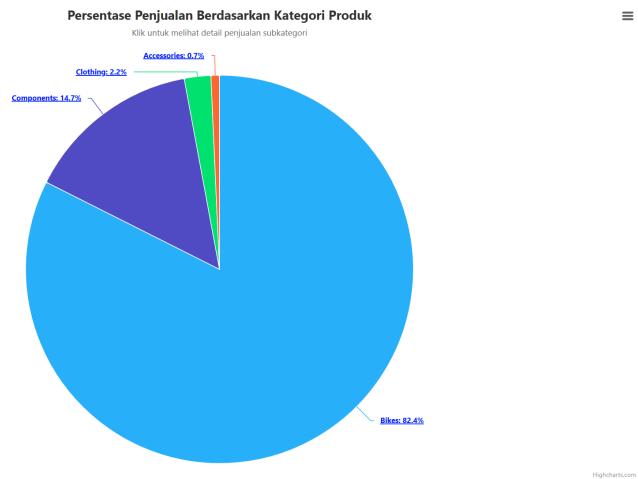
- Setelah itu lakukan run lalu tunggu. Jika berhasil, data akan otomatis masuk ke database dan dapat dilihat di Dbeaver.

factid	ProductID	Tivoli	ScrapReasonID	planedcost	ActualCost
1	517	80	1	49	49
2	517	80	1	49	49
3	518	101	1	36.75	36.75
4	518	141	1	49	49
5	748	445	1	92.25	92.25
6	748	445	1	87.5	87.5
7	748	445	1	14.5	14.5
8	748	445	1	49	49
9	748	445	1	36.75	36.75
10	748	445	1	49	49
11	994	537	1	36.75	36.75
12	763	637	1	49	49
13	798	718	1	49	49
14	804	718	1	92.25	92.25
15	804	718	1	36.75	36.75
16	959	932	1	49	49
17	819	968	1	36.75	36.75
18	793	998	1	49	49
19	977	1023	1	49	49
20	945	1023	1	92.25	92.25
21	945	1023	1	36.75	36.75
22	803	1115	2	36.75	36.75
23	802	1115	2	36.75	36.75

Gambar 2.25 Data masuk ke factproduct

2.3 Implementasi Dashboard

a. Diagram drill down persentase penjualan berdasarkan kategori produk

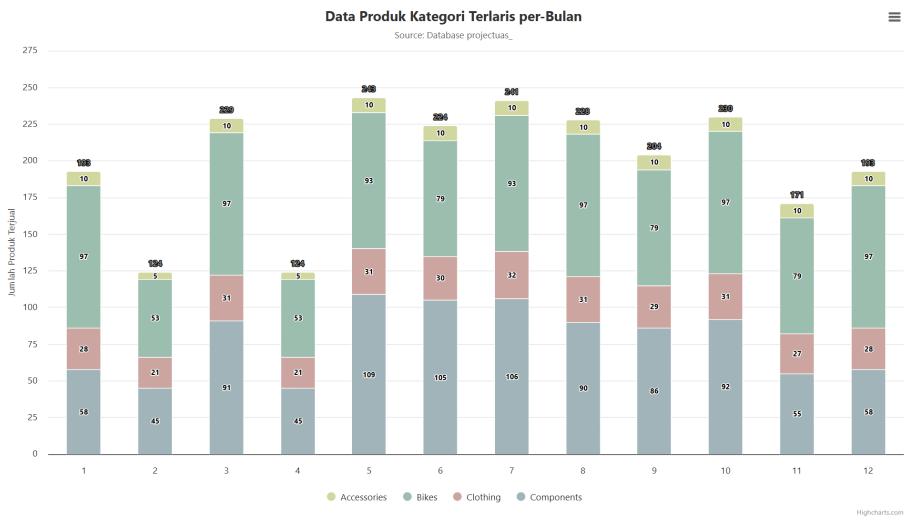


Gambar 2.26 Diagram drill down persentase penjualan kategori produk

Diagram ini memberikan gambaran visual yang jelas tentang bagaimana kontribusi tiap kategori produk terhadap total penjualan yang dihasilkan. Contohnya, jika kategori "Bikes" menyumbang 82,4% dari total penjualan, ini menunjukkan bahwa kategori tersebut memegang peran yang sangat signifikan dalam pendapatan perusahaan. Dengan menggunakan visualisasi berupa drill down chart, pengguna dapat dengan mudah melihat kategori mana yang memberikan kontribusi terbesar dan mana yang membutuhkan perhatian lebih dalam pengembangan atau pemasaran.

Informasi ini dapat digunakan untuk menentukan prioritas dalam pengalokasian anggaran pemasaran, pengembangan produk baru, serta mengoptimalkan manajemen inventaris. Jika ada kategori yang kurang memberikan kontribusi, perusahaan bisa merancang strategi untuk meningkatkan penjualannya, seperti melakukan promosi khusus atau memperkenalkan inovasi produk.

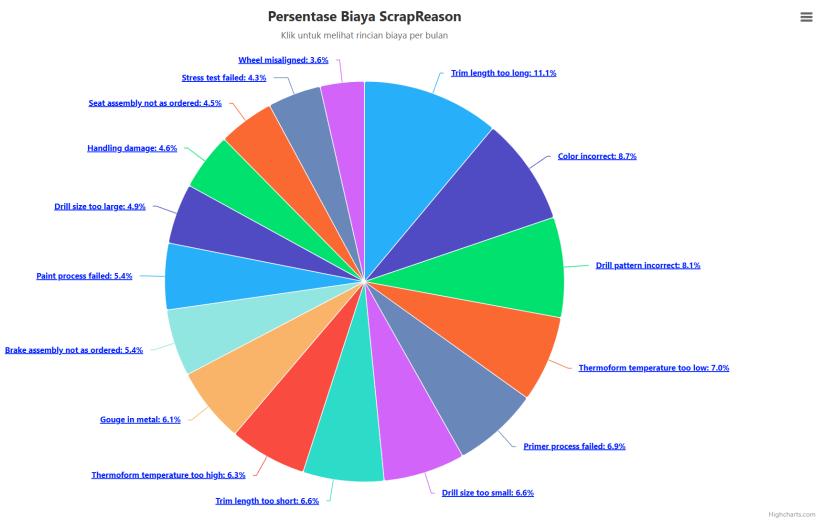
b. Diagram produk kategori terlaris per-bulan



Gambar 2.27 Diagram produk kategori terlaris per-bulan

Diagram ini menampilkan data penjualan dari produk kategori yang paling laris setiap bulan, memberikan wawasan yang lebih dalam mengenai fluktuasi penjualan sepanjang waktu. Sebagai contoh, jika produk tertentu mengalami lonjakan penjualan pada bulan tertentu, ini dapat mengindikasikan keberhasilan kampanye pemasaran atau perubahan preferensi konsumen yang perlu diperhatikan. Dengan menganalisis data ini, perusahaan bisa mengidentifikasi tren musiman, pola perubahan dalam perilaku konsumen, serta produk yang paling diminati dalam periode tertentu. Hal ini sangat berguna untuk merencanakan ketersediaan stok barang secara lebih akurat dan menyusun strategi promosi yang lebih efektif. Di samping itu, informasi ini juga memungkinkan perusahaan untuk menilai apakah ada produk yang harus lebih sering dipromosikan, atau produk yang perlu dikeluarkan dari penawaran karena kurang diminati.

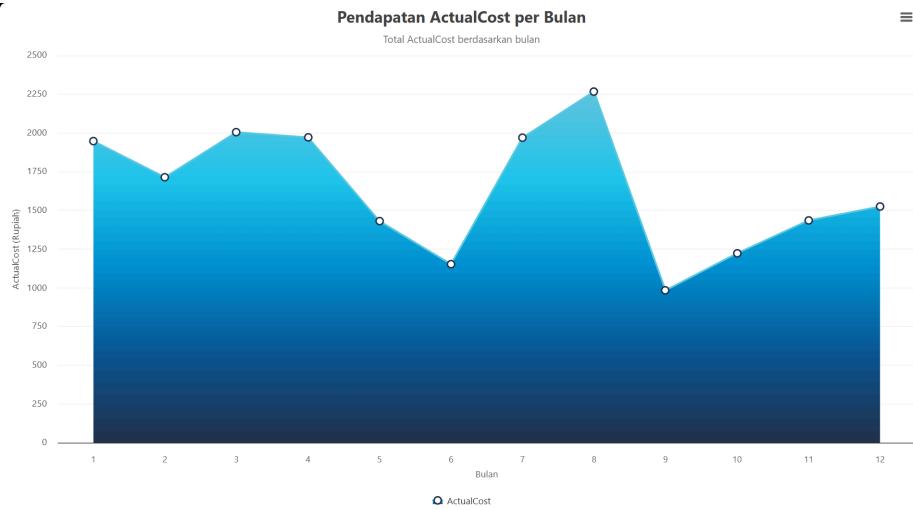
c. Diagram drill down persentase biaya scrap reason



Gambar 2.28 Diagram drill down persentase biaya scrap reason

Diagram ini memungkinkan analisis lebih mendalam tentang alasan mengapa terjadi biaya scrap (produk yang terbuang atau rusak), memberikan wawasan tentang faktor-faktor yang menyebabkan kerugian bagi perusahaan. Misalnya, jika 8,7% dari biaya scrap disebabkan oleh masalah dalam "Warna Salah," ini menunjukkan bahwa ada kesalahan dalam proses pewarnaan atau pengendalian kualitas warna yang menyebabkan produk tidak memenuhi standar yang diinginkan. Persentase ini bisa menjadi indikator penting bagi perusahaan untuk fokus pada perbaikan dalam proses produksi atau inspeksi kualitas yang berkaitan dengan warna produk. Dengan mengidentifikasi masalah ini, perusahaan bisa memperbaiki teknik pewarnaan, memeriksa bahan baku yang digunakan, atau meningkatkan kontrol kualitas untuk mengurangi tingkat scrap dan mengoptimalkan efisiensi produksi, yang pada akhirnya akan mengurangi biaya dan meningkatkan profitabilitas.

d. Diagram pendapatan actual cost per-bulan



Gambar 2.29 Diagram pendapatan actual cost per-bulan

Diagram ini menampilkan biaya aktual yang dikeluarkan perusahaan setiap bulan. Fungsinya adalah untuk memberikan gambaran yang jelas tentang jumlah pengeluaran yang dikeluarkan pada periode tertentu, tanpa membandingkannya dengan pendapatan. Melalui grafik ini, perusahaan dapat memantau bagaimana biaya berubah setiap bulan, serta mengidentifikasi bulan dengan pengeluaran lebih tinggi atau lebih rendah.

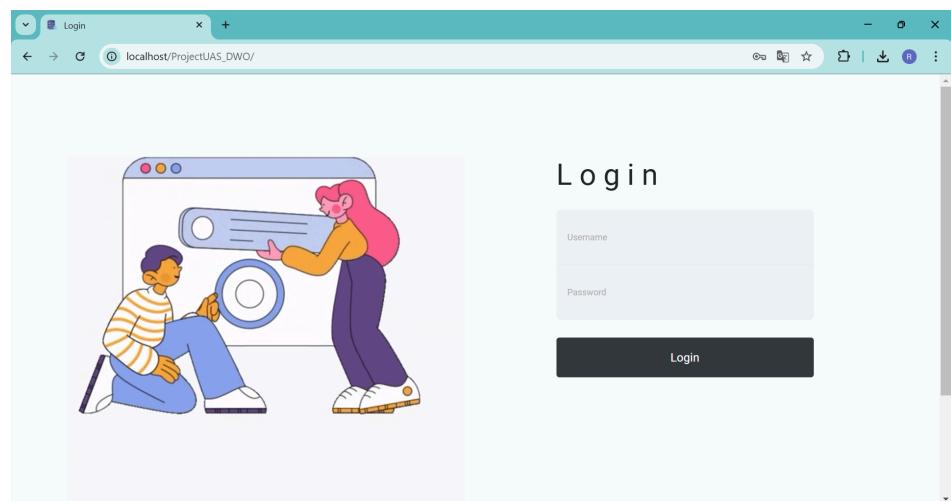
3. Lampiran

a) LINK ke GIT

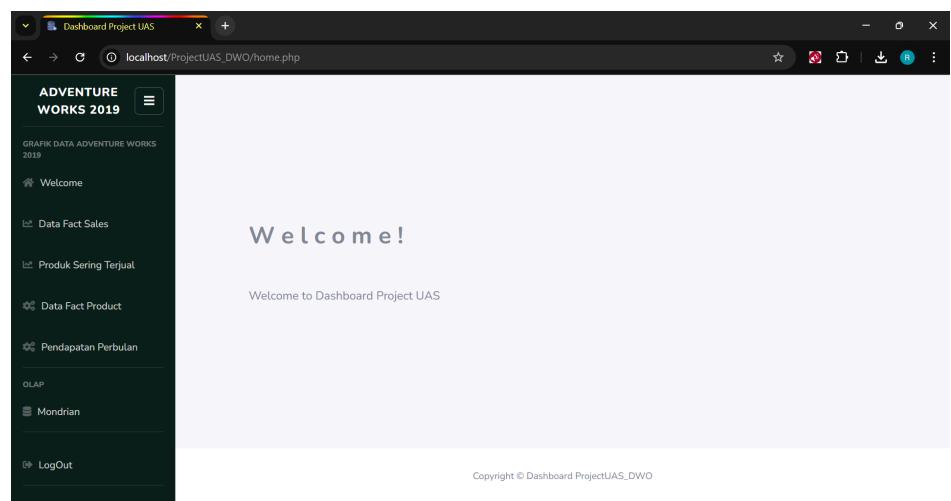
[link github Project UAS DWO](#)

b) Tampilan Dashboard

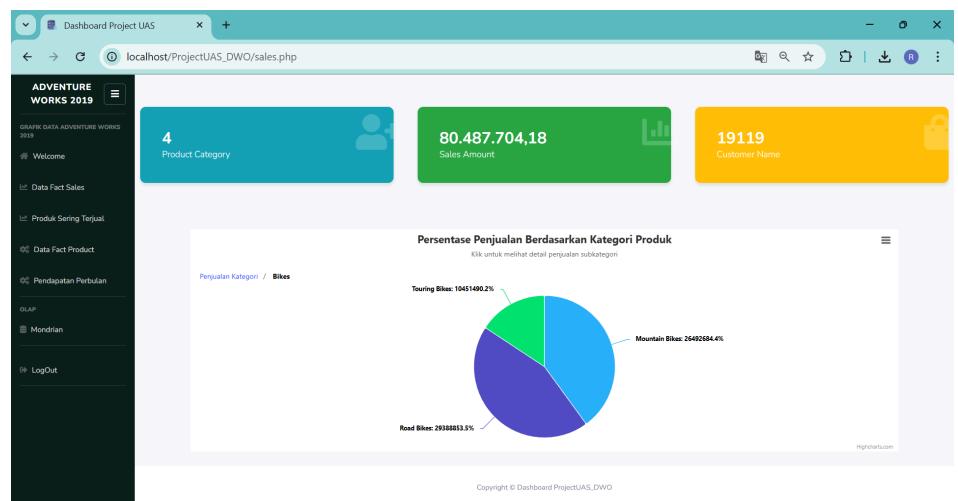
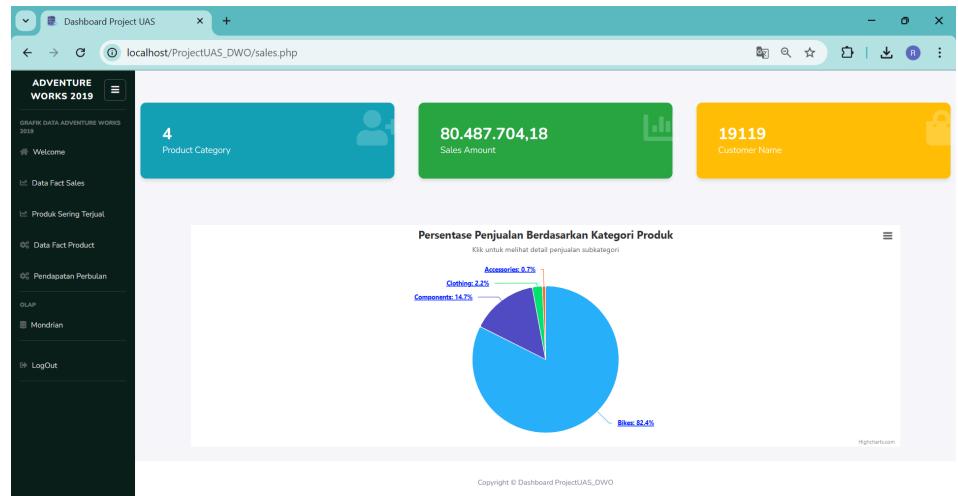
1) Halaman login



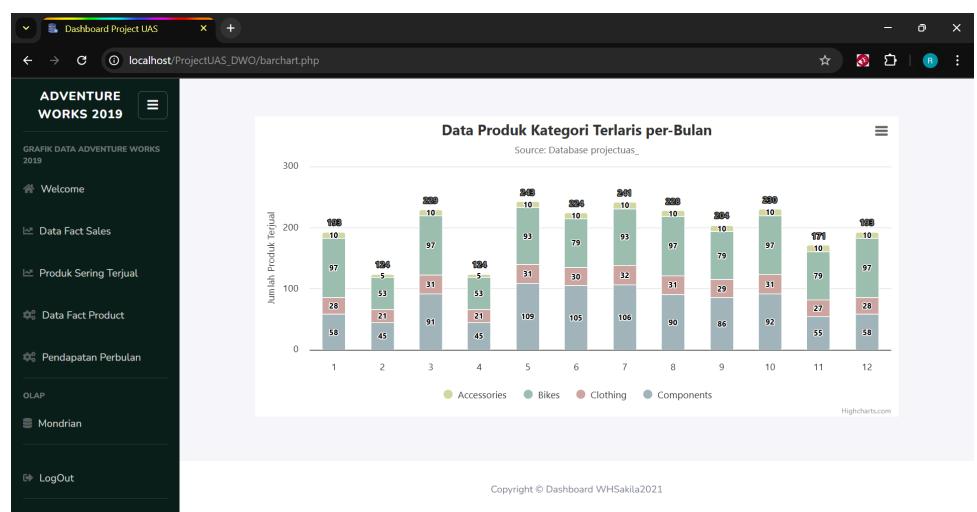
2) Halaman homepage



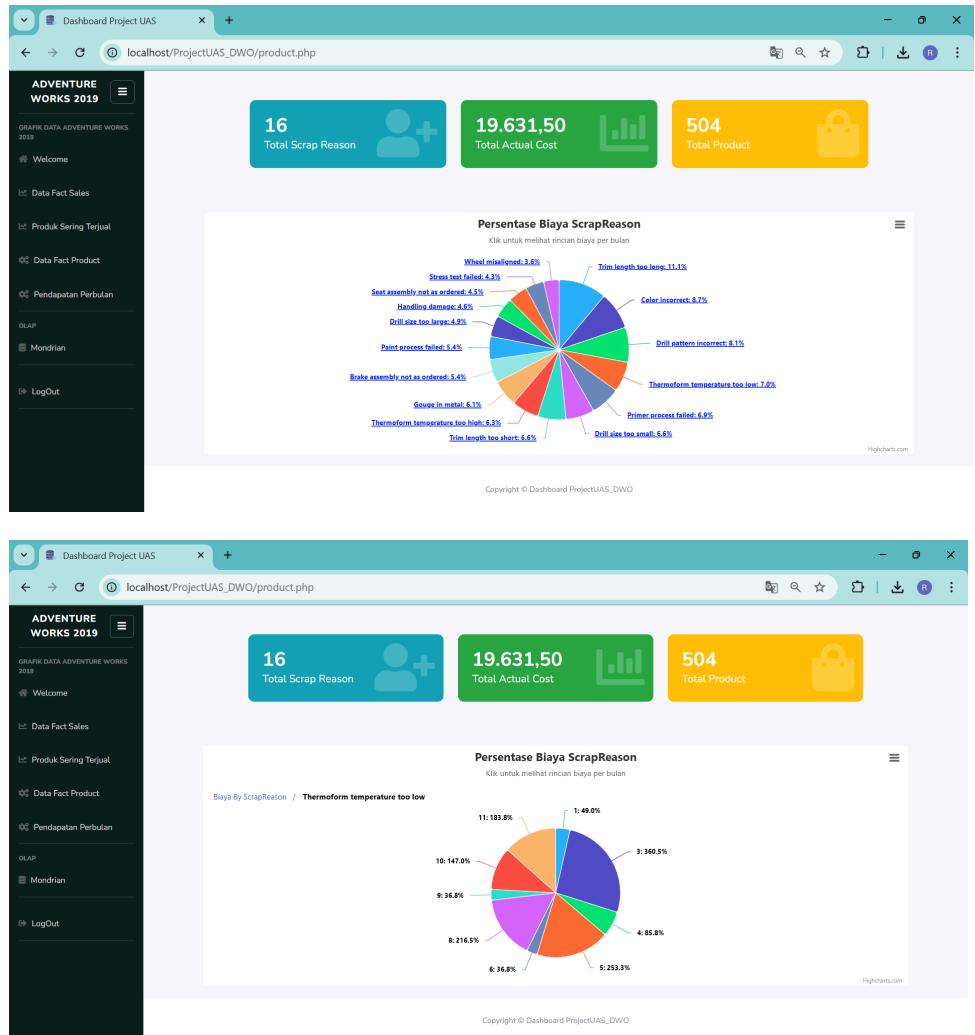
- 3) Halaman data fakta sales yaitu persentase penjualan berdasarkan kategori produk



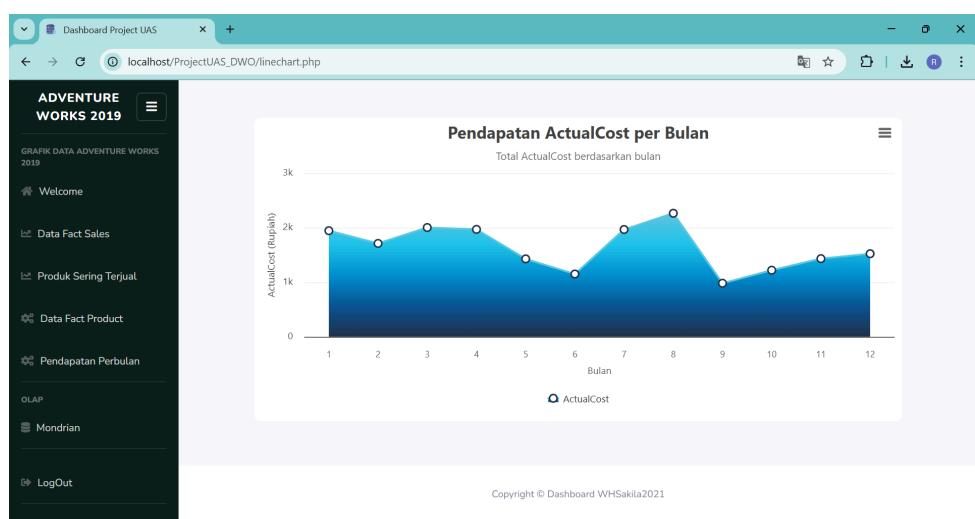
- 4) Halaman data fakta sales yaitu produk kategori terlaris perbulan



5) Halaman data fakta produk yaitu persentase biaya scrap reason



6) Halaman data fakta produk yaitu pendapatan biaya penjualan perbulan



7) Halaman OLAP dashboard yang menggunakan Mondrian

The top screenshot shows the main navigation menu on the left with options like Welcome, Data Fact Sales, Produk Sering Terjual, Data Fact Product, Pendapatan Perbulan, OLAP, Mondrian, and LogOut. On the right, there's a sidebar with 'Mondrian examples:' and a list of links, followed by 'Other links:' with more links.

The bottom screenshot shows the same navigation menu. On the right, it displays a title 'Query Project UAS DWO using Mondrian OLAP' above a pivot table interface. The pivot table has a toolbar at the top with various icons. The data grid shows measures for Sales Amount and Quantity Sold across dimensions like Time, Product, Customer, Sales Territory, and Employee. A 'Slicer' section below the grid allows filtering by All Times, All Products, All Customers, All Territories, and All Employees. The total values shown are 80,487,704 for Sales Amount and 214,516 for Quantity Sold.