

UAS DATA WAREHOUSE
RANCANGAN PROYEK ETL BERDASARKAN SKEMA BINTANG
DATASET GROCERY SALES DATABASE

Kelompok 4 / SIB 2A

Arimbi Putri Hapsari	2341760016
Karina Ika Indasa	2341760042
Khuzaima Filla Januardha	2341760078
Paudra Akbar Buana	2341760063
Reza Angelina Febriyanti	2341760015



PROGRAM STUDI D4 SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2024 / 2025

1. Cari Studi Kasus Pengelolaan Data

- Cari studi kasus pengelolaan data dalam konteks perusahaan atau organisasi pada sumber data publik online
<https://www.kaggle.com/datasets/andrexibiza/grocery-sales-dataset>
- Pilih minimal 4 kasus sebagai berikut:
 - Data Employee – atribut EmployeeID, FirstName, MiddleInitial, LastName, BirthDate, Gender, CityID, HireDate
 - Data Product – atribut ProductID, ProductName, Price, CategoryID, Class, ModifyDate, Resistant, IsAllergic, VitalityDays
 - Data Sales – atribut SalesID, SalesPersonID, CustomerID, ProductID, Quantity, Discount, TotalPrice, SalesDate, TransactionNumber
 - Data Customer – atribut CustomerID, FirstName, MiddleInitial, LastName, CityID, Address

2. Pilih Studi Kasus yang Memenuhi Kriteria

Studi kasus: data product

3. Rancang Proyek ETL Berdasarkan Skema Bintang

Tentukan tabel fakta dan tabel dimensi

- Tabel Dimensi:
 - dim_product (ProductID, ProductName, CategoryID, Resistant, IsAllergic, VitalityDays, ModifyDate)
 - dim_category (CategoryID, CategoryName)
- Tabel Fakta:
 - fact_product_vitality (ProductID, ProductName, Price, CategoryID, MinVitalityDays, MinVitality, ProductID_1, Product_Max, VitalityDays, MaxVitality)
Untuk mengetahui daya tahan produk terpanjang dan terpendek semua kategori

4. Penjelasan Langkah-Langkah

• Database

- Membuat database penjualan

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0233 seconds.)

```
CREATE DATABASE penjualan;
```

- Membuat tabel dim_category

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0353 seconds.)

```
CREATE TABLE dim_category ( CategoryID INT PRIMARY KEY, CategoryName VARCHAR(50) );
```

- Membuat tabel dim_product

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0640 seconds.)

```
CREATE TABLE dim_product ( ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Price DECIMAL(10, 4), CategoryID INT, Class VARCHAR(50), ModifyDate DATETIME, Resistant VARCHAR(50), IsAllergic BIT, VitalityDays DECIMAL(5,1), FOREIGN KEY (CategoryID) REFERENCES dim_category(CategoryID) );
```

- Membuat tabel fact_product_vitality

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0238 seconds.)

```
CREATE TABLE fact_product_vitality ( ProductID INT, Product_Min VARCHAR(255), CategoryID INT, MinVitalityDays DECIMAL(6,2), MinVitality DECIMAL(6,2), ProductID_1 INT, Product_Max VARCHAR(255), MaxVitalityDays DECIMAL(6,2), MaxVitality DECIMAL(6,2), CONSTRAINT fk_fact_product_min FOREIGN KEY (ProductID) REFERENCES dim_product(ProductID), CONSTRAINT fk_fact_product_max FOREIGN KEY (ProductID_1) REFERENCES dim_product(ProductID), CONSTRAINT fk_fact_category FOREIGN KEY (CategoryID) REFERENCES dim_category(CategoryID) );
```

• Dim_category

- Buat transformasi dim_category dengan desain seperti pada gambar



Tools yang digunakan: Input CSV file, Select Values, Filter Rows, Table Output

- Pada Input CSV file ubah step name = category_csv dan impor file categories.csv yang telah didownload. Setelah itu tekan get field dan simpan

CSV file input

Step name: Category_csv

Filename: D:\DataWarehouse\UAS\categories.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim typ
1	CategoryID	Integer	#	15	0	IDR	.	,	none
2	CategoryName	String		11		IDR	.	,	none

- Pada Select Values bagian tab select & alter klik get field lalu simpan

Select values

Step name: Select values

Select & Alter Remove Meta-data

Fields:

#	Fieldname	Rename to	Length	Precision
1	CategoryID			
2	CategoryName			

Get fields to select

Edit Mapping

- Pada filter rows atur seperti tampilan berikut agar semua field tidak ada inputan yang kosong

Filter rows

Step name: Filter rows

Send 'true' data to step: Table output

Send 'false' data to step:

The condition:

☐ CategoryID IS NOT NULL

AND

☐ CategoryName IS NOT NULL

- Terakhir table output, buat koneksi ke database penjualan. Lalu isikan target schema dan target table seperti dibawah ini. Kemudian klik simpan

Table output

Step name: Table output

Connection: dw_penjualan

Target schema: penjualan

Target table: dim_category

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	CategoryID	CategoryID
2	CategoryName	CategoryName

Get fields

Enter field mapping

- Run transformasi dim_category dan inputan akan masuk ke tabel dim_category yang telah dibuat sebelumnya pada database

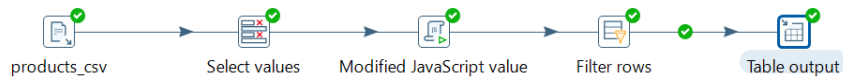
Showing rows 0 - 10 (11 total, Query took 0.0005 seconds.)

```
SELECT * FROM `dim_category`
```

CategoryID	CategoryName
1	Confections
2	Shell fish
3	Cereals
4	Dairy
5	Beverages

Dim_product

- Buat transformasi dim_product dengan desain seperti pada gambar



Tools yang digunakan: Input CSV file, Select Values, Modified JavaScript Value, Filter Rows, Table Output

- Pada Input CSV file ubah step name = product_csv dan impor file products.csv yang telah didownload. Setelah itu tekan get field dan simpan

CSV file input

Step name: products_csv

Filename: D:/DataWarehouse/UAS/products.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion?: ☒

Header row present?: ☒

Add filename to result: ☐

The row number field name (optional):

Running in parallel?: ☐

New line possible in fields?: ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal
1	ProductID	Integer	#	15	0	IDR	.
2	ProductName	String		26		IDR	.
3	Price	Number	#,.	7	4	IDR	.
4	CategoryID	Integer	#	15	0	IDR	.
5	Class	String		6		IDR	.
6	ModifyDate	Date	yyyy-MM-dd HH:mm:ss.SSS			IDR	.
7	Resistant	String		7		IDR	.
8	IsAllergic	String		7		IDR	.
9	VitalityDays	Number	#,.	5	1	IDR	.

- Pada Select Values bagian tab select & alter klik get field lalu simpan

#	Fieldname	Rename to	Length	Precision
1	ProductID			
2	ProductName			
3	Price			
4	CategoryID			
5	Class			
6	ModifyDate			
7	Resistant			
8	IsAllergic			
9	VitalityDays			

- Pada Modified JavaScript Value ketikkan kode program seperti berikut dikolom script1

Java script :

```
Script 1
is_allergic_num = (IsAllergic == "True") ? 1 : 0;
```

Lalu pada kolom field tambahkan field baru bernama is_allergic_num untuk mengubah tipe data semula string menjadi integer dan klik simpan

Fields						
#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	is_allergic_num		Integer			N

- Pada filter rows atur seperti tampilan berikut agar semua field tidak ada inputan yang kosong

The condition:

ProductID IS NOT NULL

AND

ProductName IS NOT NULL

AND

Price IS NOT NULL

AND

CategoryID IS NOT NULL

AND

Class IS NOT NULL

AND

ModifyDate IS NOT NULL

AND

Resistant IS NOT NULL

AND

VitalityDays IS NOT NULL

AND

is_allergic_num IS NOT NULL

- Terakhir table output, buat koneksi ke database penjualan. Lalu isikan target schema dan target table serta ubah Field to insert point 8 seperti dibawah ini. Kemudian klik simpan

#	Table field	Stream field
1	ProductID	ProductID
2	ProductN...	ProductName
3	Price	Price
4	CategoryID	CategoryID
5	Class	Class
6	ModifyDate	ModifyDate
7	Resistant	Resistant
8	IsAllergic	is_allergic_p...
9	VitalityDays	VitalityDays

- Run transformasi dim_product dan inputan akan masuk ke tabel dim_product yang telah dibuat sebelumnya pada database

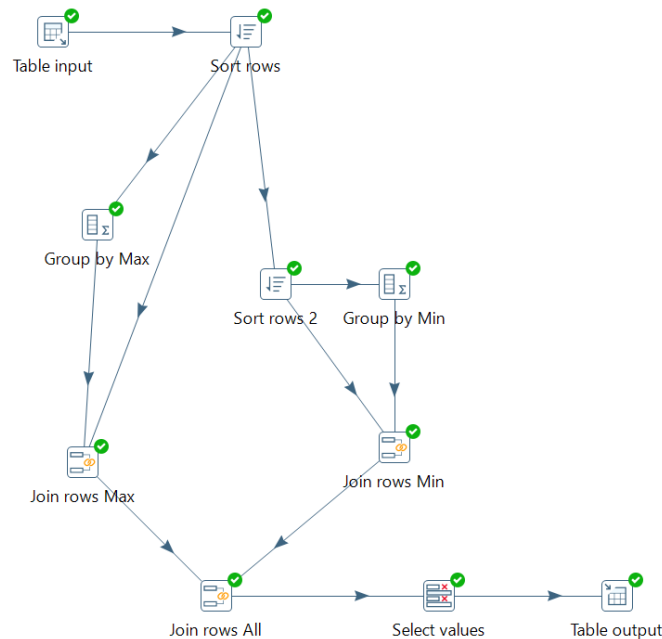
✓ Showing rows 0 - 24 (452 total, Query took 0.0065 seconds.)

SELECT * FROM `dim_product`

ProductID	ProductName	Price	CategoryID	Class	ModifyDate	Resistant	IsAllergic	VitalityDays
1	Flour - Whole Wheat	74.2988	3	Medium	2018-02-16 08:21:49	Durable	0	0.0
2	Cookie Chocolate Chip With	91.2329	3	Medium	2017-02-12 11:39:11	Unknown	0	0.0
3	Onions - Cipolini	9.1379	9	Medium	2018-03-15 08:11:52	Weak	0	111.0
4	Sauce - Gravy, Au Jus, Mix	54.3055	9	Medium	2017-07-16 00:46:29	Durable	0	0.0
5	Artichokes - Jerusalem	65.4771	2	Low	2017-08-16 14:13:35	Durable	1	27.0

- **Fact_product_vitality**

- Buat transformasi fact_product_vitality seperti gambar



- Pada Table Input masukkan query seperti ini untuk menghubungkan ke tabel dim_product

Table input

Step name: Table input

Connection: dw_penjualan

SQL: `SELECT ProductID, ProductName, CategoryID, VitalityDays FROM dim_product;`

- Pada Sort rows atur CategoryID agar diurutkan secara ASC

Sort rows

Step name: Sort rows

Sort directory: %%java.io.tmpdir%%

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	CategoryID	Y	N	N	0	N

- Pada Group by Max atur seperti berikut untuk menampilkan VitalityProduct yang paling lama untuk masing-masing kategori

Group by

Step name: Group by Max

Include all rows? ☐

Temporary files directory: %%java.io.tmpdir%% [Browse...](#)

TMP-file prefix: grp

Add line number, restart in each group ☐

Line number field name:

Always give back a result row ☐

The fields that make up the group:

#	Group field
1	CategoryID

Get Fields

Aggregates :

#	Name	Subject	Type	Value
1	MaxVitality	VitalityDays	Maximum	

Get lookup fields

- Pada Join rows max atur seperti berikut untuk menggabungkan data semula dengan agregasi yang dilakukan pada tahap sebelumnya

Join rows (cartesian product)

Step name: Join rows Max

Temp directory: %%java.io.tmpdir%%

TMP-file prefix: out

Max. cache size (in rows): 500

Main step to read from: Sort rows

The condition:

☐

CategoryID = CategoryID

AND

VitalityDays = MaxVitality

- Pada Sort rows 2 atur CategoryID agar diurutkan secara ASC

Sort rows

Step name: Sort rows 2

Sort directory: %%java.io.tmpdir%% [Browse...](#)

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	CategoryID	Y	N	N	0	N

- Pada Group by Min atur seperti berikut untuk menampilkan VitalityProduct yang paling cepat untuk masing-masing kategori

Group by

Step name: Group by Min

Include all rows? ☐

Temporary files directory: %%java.io.tmpdir%% [Browse...](#)

TMP-file prefix: grp

Add line number, restart in each group ☐

Line number field name:

Always give back a result row ☐

The fields that make up the group:

#	Group field
1	CategoryID

Get Fields

Aggregates :

#	Name	Subject	Type	Value
1	MinVitality	VitalityDays	Minimum	

Get lookup fields

- Pada Join rows min atur seperti berikut untuk menggabungkan data semula dengan agregasi yang dilakukan pada tahap sebelumnya

Join rows (cartesian product)

Step name: Join rows Min

Temp directory: %%java.io.tmpdir%%

TMP-file prefix: out

Max. cache size (in rows): 500

Main step to read from: Sort rows 2

The condition:

☐ CategoryID = CategoryID

AND

VitalityDays = MinVitality

- Pada join rows all atur seperti berikut untuk menggabungkan hasil agregasi min dan max sebelumnya

Join rows (cartesian product)

Step name: Join rows All

Temp directory: %%java.io.tmpdir%%

TMP-file prefix: out

Max. cache size (in rows): 500

Main step to read from: Join rows Min

The condition:

☐ CategoryID = CategoryID_1

- Pada select values pilih field yang hendak digunakan serta rename

Select values

Step name: Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	ProductID			
2	ProductName	Product_Min		
3	CategoryID			
4	VitalityDays	MinVitalityDays		
5	CategoryID_1			
6	MinVitality			
7	ProductID_1			
8	ProductName_1	Product_Max		
9	CategoryID_2			
1.	VitalityDays_1	MaxVitalityDays		
1.	CategoryID_1_1			
1.	MaxVitality			

Hapus field yang tidak perlu berikut

Select values

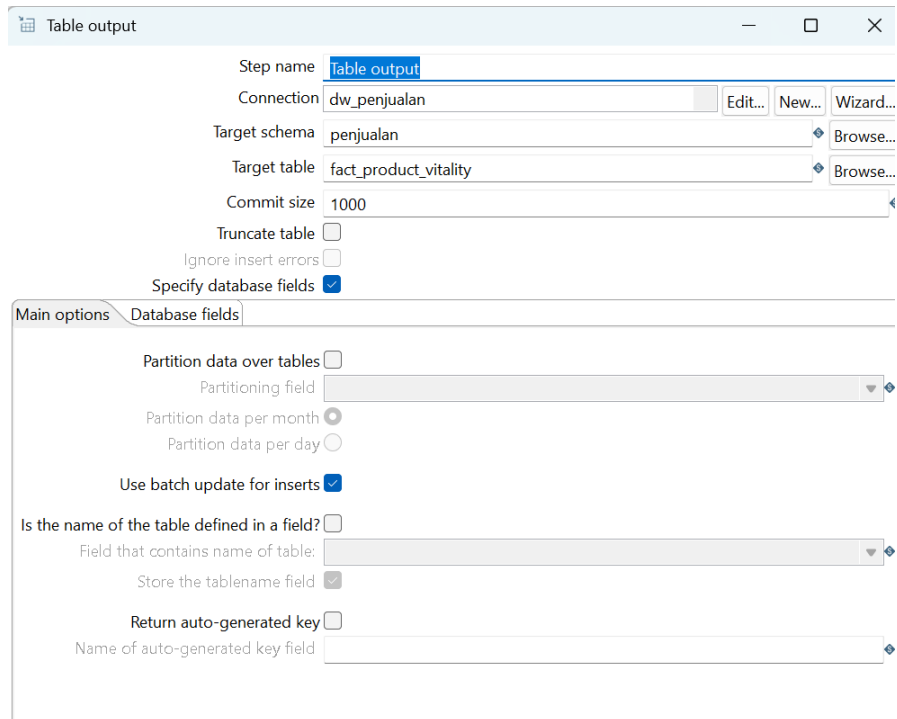
Step name: Select values

Select & Alter Remove Meta-data

Fields to remove :

#	Fieldname
1	CategoryID_1
2	CategoryID_2
3	CategoryID_1_1

- Pada table output hubungkan ke database serta inputkan target table sebagai berikut



- Run transformasi fact_vitality_product dan inputan akan masuk ke tabel fact_vitality_product yang telah dibuat sebelumnya pada database

✓ Showing rows 0 - 24 (440 total, Query took 0.0003 seconds.)

```
SELECT * FROM `fact_product_vitality`
```

ProductID	Product_Min	CategoryID	MinVitalityDays	MinVitality	ProductID_1	Product_Max	MaxVitalityDays	MaxVitality
105	Ecolab - Mikroklene 4/4 L	1	0.00	0.00	8	Halibut - Steaks	108.00	108.00
105	Ecolab - Mikroklene 4/4 L	1	0.00	0.00	8	Halibut - Steaks	108.00	108.00
105	Ecolab - Mikroklene 4/4 L	1	0.00	0.00	163	Tomatoes Tear Drop	116.00	116.00
105	Ecolab - Mikroklene 4/4 L	1	0.00	0.00	163	Tomatoes Tear Drop	116.00	116.00
105	Ecolab - Mikroklene 4/4 L	1	0.00	0.00	113	Ocean Spray - Kiwi Strawberry	106.00	106.00