

**PROJECT UJIAN AKHIR SEMESTER BUSINESS INTELLIGENCE
SISTEM MANAJEMEN RUMAH SAKIT**



Disusun Oleh:

Abdullah Shamil Basayev	2341720166
Keisya Nisrina Aulia	2341720146
Muhammad Naufal Pratomo	2341720075
Rensi Meila Yulvinata	2341720201
Yefta Octavianus Santo	2341720110

Dosen Pengampu:

Endah Septa Sintya, S.Pd, M.Kom

**JURUSAN TEKNOLOGI INFORMASI
D4 TEKNIK INFORMATIKA
POLITEKNIK NEGERI MALANG
2025**

LINK GITHUB

<https://github.com/rensimeila04/hims-bussiness-intelligence>

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Industri layanan kesehatan modern menghasilkan volume data yang sangat besar setiap harinya. Rumah sakit, sebagai instansi pelayanan publik yang krusial, mencatat ribuan transaksi mulai dari pendaftaran pasien, penjadwalan dokter, tindakan medis, hingga proses penagihan biaya layanan. Data ini merupakan aset berharga yang, jika dikelola dengan benar, dapat memberikan wawasan strategis untuk peningkatan efisiensi operasional dan kualitas layanan.

Saat ini, sistem manajemen rumah sakit (HIMS) yang berjalan menggunakan basis data operasional atau *Online Transaction Processing* (OLTP). Struktur data ini dirancang untuk kecepatan transaksi harian (*input-oriented*). Namun, struktur ini memiliki kelemahan signifikan ketika digunakan untuk kebutuhan analisis dan pelaporan (*read-oriented*), seperti sulitnya memantau performa keuangan secara holistik karena data tersebar di tabel yang terfragmentasi.

Oleh karena itu, diperlukan solusi *Business Intelligence* (BI) melalui pembangunan *Data Warehouse* untuk memisahkan beban kerja analisis dan menyajikan data terintegrasi.

1.2 Identifikasi Masalah

Berdasarkan analisis sistem berjalan, terdapat kendala utama, yaitu Fragmentasi Data. Data tersimpan terpisah (pasien, dokter, transaksi), menyulitkan pandangan utuh.

Kinerja Pelaporan Query analitik membebani database operasional.

Kualitas Data Format data tidak seragam (misal: nama terpisah depan/belakang) dan format tanggal yang kaku.

1.3 Rumusan Masalah

1. Bagaimana mengintegrasikan data HIMS yang terpisah menjadi satu pusat data?
2. Bagaimana merancang arsitektur *Data Warehouse* yang mendukung analisis cepat?
3. Bagaimana menerapkan ETL (*Extract, Transform, Load*) menggunakan Pentaho untuk standarisasi data?

1.4 Tujuan

1. Membangun *Data Warehouse* dengan model dimensi.
2. Mengimplementasikan ETL menggunakan Pentaho Data Integration (PDI).
3. Menyajikan data yang telah diperkaya (*enriched*) untuk laporan pendapatan.

BAB 2

ANALISIS SUMBER DATA DAN SISTEM

2.1 Tinjauan Sistem Sumber (OLTP)

Data bersumber dari HIMS_Dataset.sql (MySQL). Struktur ini sangat ternormalisasi. Tabel utama yang diidentifikasi:

- patients (Data demografi)
- doctors (Data medis)
- hospital_charges (Data transaksi/fakta utama)

Server: 127.0.0.1 -> Database: hms													
Structure	SQL	Search	Query	Export	Import	Operations	Privileges	Routines	Events	Triggers			
<input type="checkbox"/> accessrights	★							10	InnoDB	utf8mb4_0900_ai_ci	48.0	KiB	-
<input type="checkbox"/> admission	★							3,996	InnoDB	utf8mb4_0900_ai_ci	1.4	MiB	-
<input type="checkbox"/> advance	★							0	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> applicationfunctionality	★							6	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> bed	★							50	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> bill	★							12,050	InnoDB	utf8mb4_0900_ai_ci	2.0	MiB	-
<input type="checkbox"/> charge	★							13,669	InnoDB	utf8mb4_0900_ai_ci	2.7	MiB	-
<input type="checkbox"/> city	★							5	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> clinicalnote	★							1	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> country	★							1	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> department	★							24	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> discharge	★							3,996	InnoDB	utf8mb4_0900_ai_ci	576.0	KiB	-
<input type="checkbox"/> doctor	★							60	InnoDB	utf8mb4_0900_ai_ci	64.0	KiB	-
<input type="checkbox"/> gender	★							2	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> insurancecompany	★							7	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> nationality	★							1	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> patient	★							~53,459	InnoDB	utf8mb4_0900_ai_ci	13.5	MiB	-
<input type="checkbox"/> patientcategory	★							4	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> person	★							8	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> prescription	★							1	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> relation	★							14	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> room	★							13	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> salesserver1	★							0	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> service	★							321	InnoDB	utf8mb4_0900_ai_ci	112.0	KiB	-
<input type="checkbox"/> servicecategory	★							15	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> servicetype	★							3	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> staff	★							120	InnoDB	utf8mb4_0900_ai_ci	96.0	KiB	-
<input type="checkbox"/> state	★							1	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> sysdiagrams	★							2	InnoDB	utf8mb4_0900_ai_ci	208.0	KiB	-
<input type="checkbox"/> temppatient	★							9,966	InnoDB	utf8mb4_0900_ai_ci	1.5	MiB	-
<input type="checkbox"/> temptpt	★							2	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> unit	★							5	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
<input type="checkbox"/> user	★							3	InnoDB	utf8mb4_0900_ai_ci	32.0	KiB	-
<input type="checkbox"/> visit	★							10,815	InnoDB	utf8mb4_0900_ai_ci	2.7	MiB	-
<input type="checkbox"/> ward	★							7	InnoDB	utf8mb4_0900_ai_ci	16.0	KiB	-
35 tables	Sum							~108,637	InnoDB	utf8mb4_0900_ai_ci	25.5	MiB	0 B

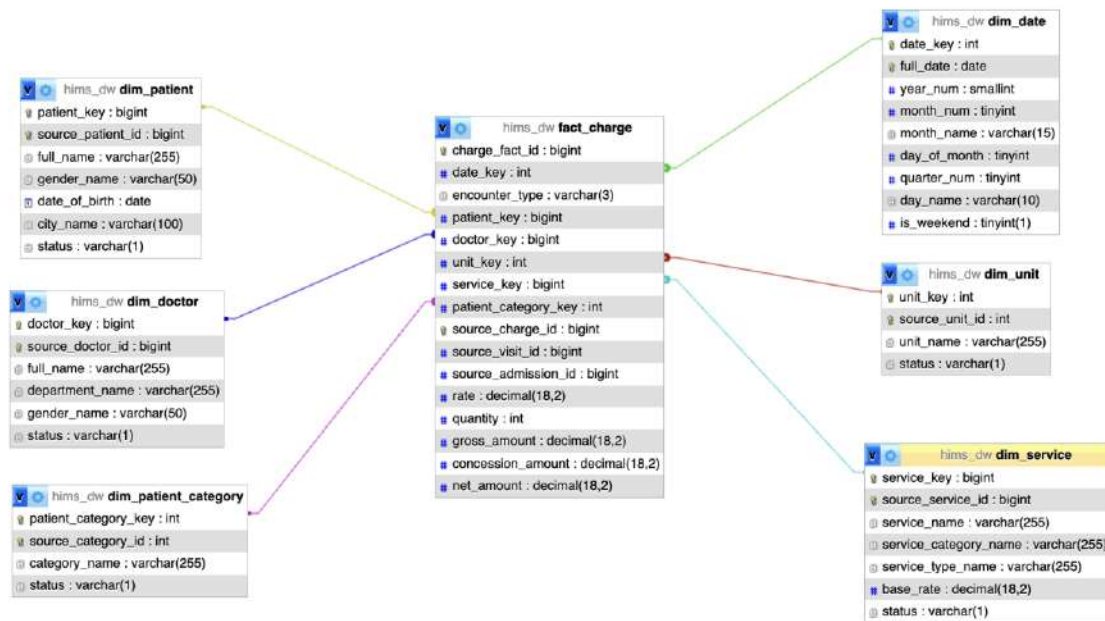
2.2 Perancangan Skema Bintang (Star Schema)

Sistem BI ini dirancang menggunakan *Star Schema* dengan tabel fakta di tengah.

Entitas Data Warehouse:

1. **Fact Table:** fact_charge (Menyimpan metrik biaya, gross, net, quantity).
2. **Dimension Tables:** dim_patient, dim_doctor, dim_unit, dim_service, dim_patient_category.
3. **Time Dimension:** dim_date (Dibuat khusus untuk analisis temporal).

Diagram Star Schema OLAP



BAB 3

IMPLEMENTASI DATABASE DAN DIMENSI WAKTU

Sebelum proses ETL, struktur tabel tujuan disiapkan di database hims_dw. Tabel fakta disiapkan dengan *Foreign Key* yang mengarah ke tabel dimensi.

Table	Action
<input type="checkbox"/> dim_date	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> dim_doctor	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> dim_patient	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> dim_patient_category	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> dim_service	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> dim_unit	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> fact_charge	★ Browse Structure Search Insert Empty Drop

BAB 4

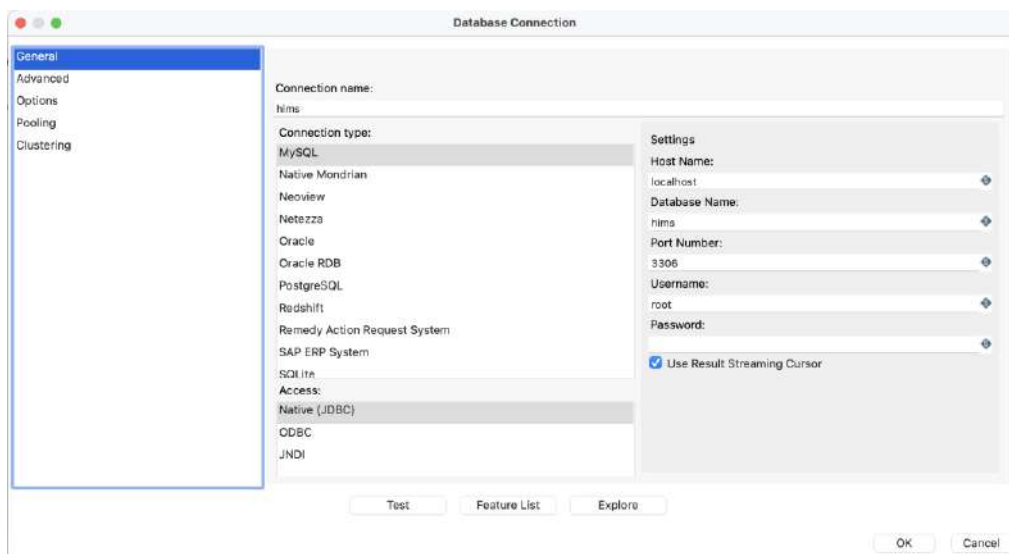
IMPLEMENTASI ETL DENGAN PENTAHO (PDI)

4.1 Konfigurasi Koneksi Database

Langkah pertama adalah memastikan Pentaho dapat berkomunikasi dengan kedua sisi basis data.

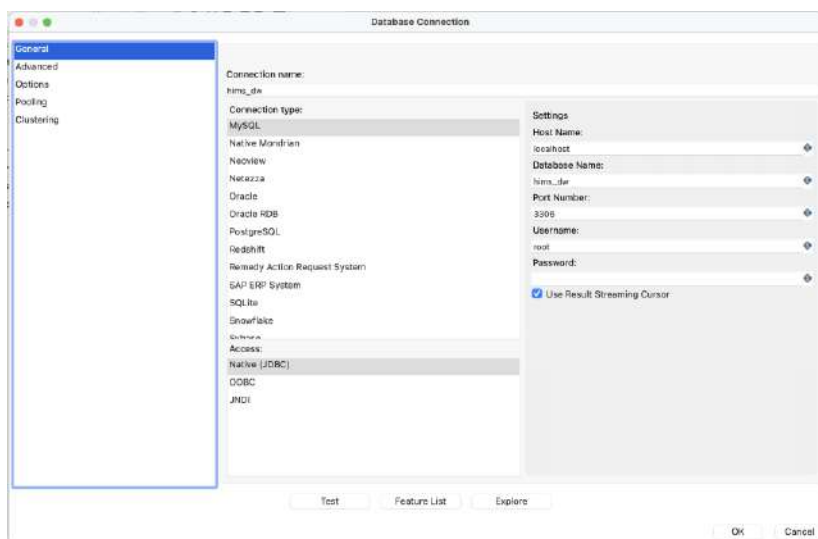
1. Koneksi Database Sumber (OLTP)

Koneksi ini menghubungkan Pentaho dengan database MySQL hims_oltp untuk mengambil data mentah.



2. Koneksi Database Tujuan (OLAP)

Koneksi ini menghubungkan Pentaho dengan database MySQL hims_dw untuk menyimpan hasil data warehouse.



4.2 Proses ETL: Load Tabel Dimensi (Contoh: dim_unit)

Setelah koneksi siap, proses pengisian tabel dimensi dilakukan. Berikut adalah langkah-langkah detail pengisian dim_unit.

a. Desain Transformasi (HIMS_dim unit)

Langkah awal adalah membuat desain transformasi pada canvas Pentaho Spoon yang menghubungkan langkah ekstraksi dan loading.

Transformation HIMS_dim unit



b. Step Extract (Pengambilan Data)

Menggunakan step *Table Input*, data diambil dari tabel sumber units pada database OLTP. Query SQL digunakan untuk menyeleksi kolom yang relevan.

Step Extract (ambil data dari OLTP)



Rows of step: Table input (5 rows)

#	source_unit_id	unit_name	status
1	1	Unit 1	Y
2	2	Unit 2	Y
3	3	Unit 3	Y
4	4	Unit 4	Y
5	5	Unit 5	Y

c. Step Load (Pemuatan Data)

Data yang telah diekstrak kemudian dimasukkan ke tabel tujuan dim_unit pada database Data Warehouse menggunakan step *Table Output* (atau *Dimension Lookup/Update*).

Insert / update

Step name: Insert / update

Connection: hims_dw

Target schema: hims_dw

Target table: dim_unit

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	source_unit_id	=	source_unit_id	

Update fields:

#	Table field	Stream field	Update
1	source_unit_id	source_unit_id	N
2	unit_name	unit_name	Y
3	status	status	Y

d. Eksekusi Transformasi (Run)

Menjalankan transformasi untuk memproses data dari sumber ke tujuan. Jendela eksekusi menampilkan status keberhasilan proses.

Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

#

Stepname

Copynr

Read

Written

Input

Output

Updated

Rejected

Errors

Active

Time

Speed (r/s)

input/output

1

Table input

0

0

5

5

0

0

0

0

Finished

0.0s

385

-

2

Insert / update

0

5

5

5

5

0

0

0

Finished

0.0s

132

-

e. Hasil di MySQL Verifikasi akhir dilakukan dengan memeriksa isi tabel dim_unit pada database MySQL hims_dw untuk memastikan data telah masuk dengan benar.

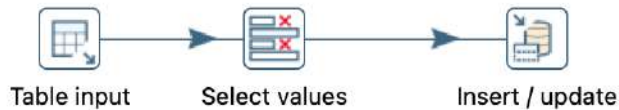
	unit_key	source_unit_id	unit_name	status
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	1	1	Unit 1	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	2	2	Unit 2	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	3	3	Unit 3	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	4	4	Unit 4	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	5	5	Unit 5	1

4.2 Proses ETL: Load Tabel Dimensi (Contoh: dim_patient_category)

Setelah koneksi siap, proses pengisian tabel dimensi dilakukan. Berikut adalah langkah-langkah detail pengisian dim_patient_category.

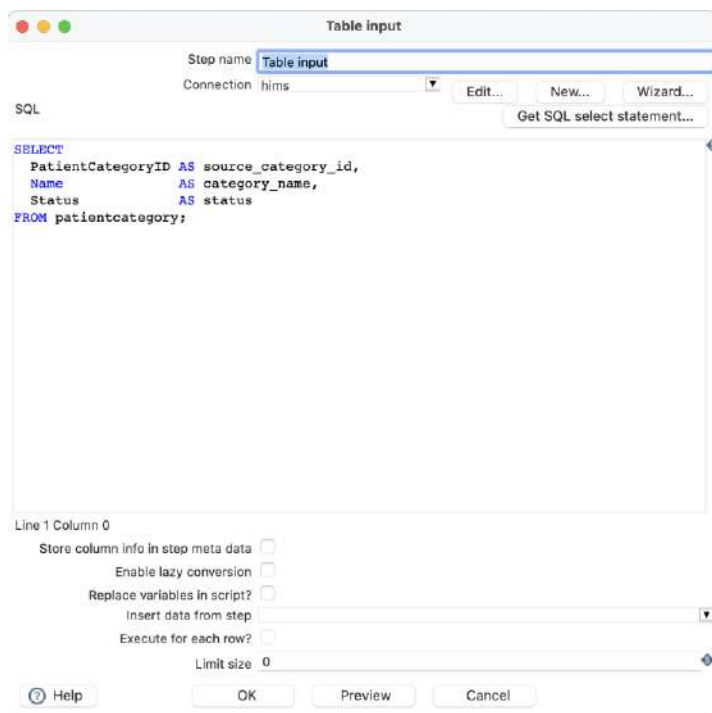
a. Desain Transformasi (HIMS_dim patient category)

Langkah awal adalah membuat desain transformasi pada canvas Pentaho Spoon yang menghubungkan langkah ekstraksi, transformasi, dan loading.



b. Step Table Input (Extract)

Menggunakan step *Table Input*, data diambil dari tabel sumber patient_categories pada database OLTP.

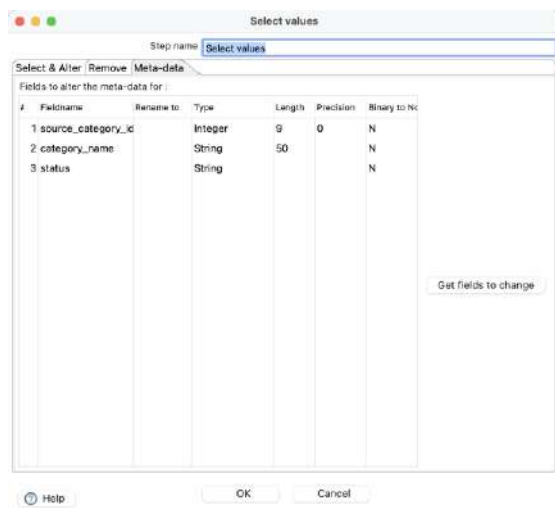


Rows of step: Table input (4 rows)

#	source_category_id	category_name	status
1		1 Self	Y
2		2 Company	Y
3		3 Staff	Y
4		4 StaffDependent	Y

c. Step Select Values

Step ini digunakan untuk memilih kolom yang relevan, mengubah nama kolom (*metadata*), atau mengatur tipe data agar sesuai dengan struktur tabel tujuan.



d. Step Insert/Update (Load)

Data kemudian dimasukkan ke tabel `dim_patient_category` menggunakan step *Insert/Update*. Step ini memastikan data baru ditambahkan dan data yang berubah diperbarui berdasarkan kunci lookup.

Insert/update

Insert / update

Step name: **Insert / update**

Connection: **hims_dw** [Edit...] [New...] [Wizard...]

Target schema: **hims_dw** [Browse...]

Target table: **dim_patient_category** [Browse...]

Commit size: **100**

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	source_category_id	=	source_category_id	

[Get fields]

Update fields:

#	Table field	Stream field	Update
1	source_category_id	source_category_id	N
2	category_name	category_name	Y
3	status	status	Y

[Get update fields] [Edit mapping]

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

e. Eksekusi Transformasi (Run)

Menjalankan transformasi untuk memproses data dari sumber ke tujuan. Jendela eksekusi menampilkan status keberhasilan proses.

Run

Execution Results

LoggingExecution HistoryStep MetricsPerformance GraphMetricsPreview data

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input	0	0	4	4	0	0	0	0	Finished	0.0s	333	-
2	Select values	0	4	4	0	0	0	0	0	Finished	0.0s	286	-
3	Insert / update	0	4	4	4	4	0	0	0	Finished	0.0s	210	-

f. Hasil di MySQL

Verifikasi akhir dilakukan dengan memeriksa isi tabel `dim_patient_category` pada database MySQL `hims_dw` untuk memastikan data telah masuk dengan benar.

Cek

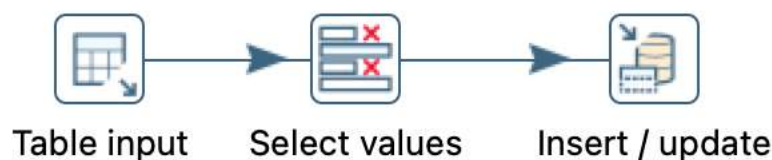
		patient_category_key	source_category_id	category_name	status
<input type="checkbox"/>	Edit Copy Delete	1	1	Self	Y
<input type="checkbox"/>	Edit Copy Delete	2	2	Company	Y
<input type="checkbox"/>	Edit Copy Delete	3	3	Staff	Y
<input type="checkbox"/>	Edit Copy Delete	4	4	StaffDependent	Y

4.3 Proses ETL: Load Tabel Dimensi Service (dim_service)

Langkah selanjutnya adalah memproses dimensi layanan (*Service*). Proses ini serupa namun menggunakan tabel sumber dan tujuan yang berbeda.

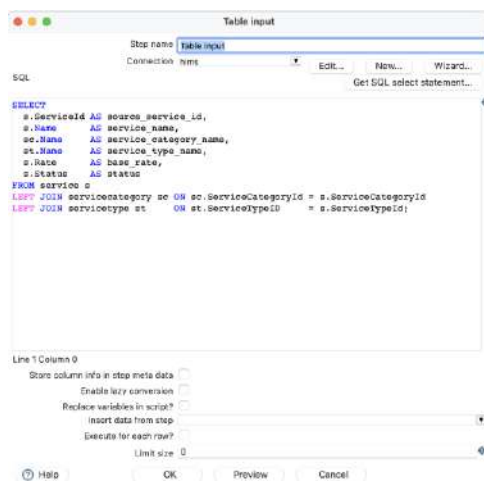
a. Desain Transformasi (HIMS_dim service)

Membuat desain alur transformasi baru untuk dimensi service.



b. Step Table Input (Extract)

Mengambil data layanan dari tabel sumber services di database OLTP.



Rows of step: Table input (321 rows)

#	source_service_id	service_name	service_category_name	service_type_name	base_rate	status
1	1	Consultation	Cardiology	OPD	200.0	Y
2	2	Consultation	Gynaecology	OPD	200.0	Y
3	3	Consultation	Medicine	OPD	200.0	Y
4	4	Consultation	Nephrology	OPD	200.0	Y
5	5	Consultation	Ophthalmology	OPD	200.0	Y
6	6	Consultation	Orthopedic	OPD	200.0	Y
7	7	Consultation	Paediatrics	OPD	200.0	Y
8	8	Consultation	Physiotherapy	OPD	200.0	Y
9	9	Consultation	Neurology	OPD	200.0	Y
10	10	Consultation	Surgery	OPD	200.0	Y
11	11	Consultation	Dental	OPD	200.0	Y
12	12	Consultation	ENT	OPD	200.0	Y
13	13	Follow-up Consultatio	Cardiology	OPD	200.0	Y
14	14	Follow-up Consultatio	Gynaecology	OPD	200.0	Y
15	15	Follow-up Consultatio	Medicine	OPD	200.0	Y
16	16	Follow-up Consultatio	Nephrology	OPD	200.0	Y
17	17	Follow-up Consultatio	Ophthalmology	OPD	200.0	Y
18	18	Follow-up Consultatio	Orthopedic	OPD	200.0	Y
19	19	Follow-up Consultatio	Paediatrics	OPD	200.0	Y
20	20	Follow-up Consultatio	Physiotherapy	OPD	200.0	Y
21	21	Follow-up Consultatio	Neurology	OPD	200.0	Y
22	22	Follow-up Consultatio	Surgery	OPD	200.0	Y
23	23	Follow-up Consultatio	Dental	OPD	200.0	Y
24	24	Follow-up Consultatio	ENT	OPD	200.0	Y
25	25	Procedure	Cardiology	OPD	200.0	Y

c. Step Select Values

Melakukan pemilihan dan pemetaan kolom seperti nama layanan (service_name) dan tarif dasar (rate).

Select

values

Step name: **Select values**

Select & Alter | Remove | Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary t
1	source_service_id		Integer	15	0	N
2	service_name		String	200		N
3	service_category_name		String	50		N
4	service_type_name		String	50		N
5	base_rate		Number	18	2	N
6	status		String			N

Get fields to change

Help OK Cancel

d. Step Insert/Update (Load)

Memasukkan data yang telah dipetakan ke dalam tabel dim_service di database tujuan.

Insert / update

Step name: **Insert / update**

Connection: **hims_dw** [Edit...] [New...] [Wizard...] [Browse...]

Target schema: [Browse...]

Target table: **dim_service** [Browse...]

Commit size: **100**

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	source_service_id	=	source_service_id	

[Get fields]

Update fields:

#	Table field	Stream field	Update
1	source_service_id	source_service_id	N
2	service_name	service_name	Y
3	service_category_name	service_category_name	Y
4	service_type_name	service_type_name	Y
5	base_rate	base_rate	Y
6	status	status	Y

[Get update fields] [Edit mapping]

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

e. Eksekusi Transformasi (Run)

Menjalankan transformasi HIMS_dim service.ktr.

Run

Execution Results													
<div> Logging Execution History Step Metrics Performance Graph Metrics Preview data </div>													
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input	0	0	321	321	0	0	0	0	Finished	0.0s	29,182	-
2	Select values	0	321	321	0	0	0	0	0	Finished	0.0s	24,692	-
3	Insert / update	0	321	321	321	321	0	0	0	Finished	0.1s	2,396	-

f. Hasil di MySQL

Memverifikasi bahwa daftar layanan rumah sakit telah masuk ke tabel dim_service di hims_dw.

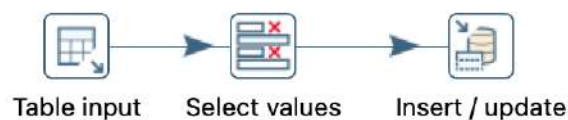
	service_key	source_service_id	service_name	service_category_name	service_type_name	base_rate	status
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Consultation	Cardiology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	Consultation	Gynaecology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	Consultation	Medicine	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	4	Consultation	Nephrology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	5	Consultation	Ophthalmology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	6	Consultation	Orthopedic	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	7	Consultation	Paediatrics	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	8	Consultation	Physiotherapy	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	9	Consultation	Neurology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	10	Consultation	Surgery	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	11	Consultation	Dental	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	12	Consultation	ENT	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	13	Follow-up Consultation	Cardiology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	14	Follow-up Consultation	Gynaecology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	15	Follow-up Consultation	Medicine	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16	16	Follow-up Consultation	Nephrology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	17	17	Follow-up Consultation	Ophthalmology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	18	18	Follow-up Consultation	Orthopedic	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	19	19	Follow-up Consultation	Paediatrics	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	20	20	Follow-up Consultation	Physiotherapy	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	21	21	Follow-up Consultation	Neurology	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	22	22	Follow-up Consultation	Surgery	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	23	23	Follow-up Consultation	Dental	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	24	24	Follow-up Consultation	ENT	OPD	200.00	Y
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	25	25	Procedure	Cardiology	OPD	200.00	Y

4.4 Proses ETL: Load Tabel Dimensi Pasien (dim_patient)

Proses ini menangani data utama pasien. Karena data nama seringkali terpisah, transformasi ini sangat penting untuk menyatukan atribut agar mudah dibaca.

a. Desain Transformasi (HIMS_dim patient)

Membuat desain alur transformasi yang menghubungkan data pasien dari sumber ke gudang data.



b. Step Table Input (Extract)

Mengambil data dari tabel patients di database OLTP.

Table input

Step name

Table input

Connection

hims

Edit...

New...

Wizard...

Get SQL select statement...

SQL

```

SELECT
  p.PatientID AS source_patient_id,
  CONCAT_WS(' ', p.FirstName, p.MiddleName, p.LastName) AS full_name,
  g.Name AS gender_name,
  p.DateOfBirth AS date_of_birth,
  c.Name AS city_name,
  p.Status AS status
FROM patient p
LEFT JOIN gender g ON g.GenderID = p.GenderID
LEFT JOIN city c ON c.CityID = p.CityID;

```

Line 1 Column 0

Store column info in step meta data

Enable lazy conversion

Replace variables in script?

Insert data from step

Execute for each row?

Limit size 0

Help

OK

Preview

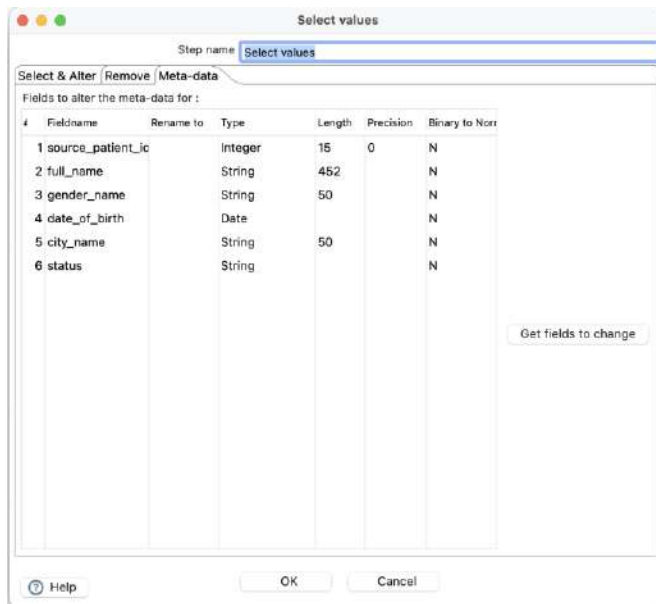
Cancel

Rows of step: Table input (1000 rows)

#	source_patient_id	full_name	gender_name	date_of_birth	city_name	status
1	1	PFname_1 PMname_1 PLname_1	Male	1995/08/29 00:00:00.00	Aurangabad	Y
2	2	PFname_2 PMname_2 PLname_2	Female	1995/08/20 00:00:00.00	Mumbai	Y
3	3	PFname_3 PMname_3 PLname_3	Male	1995/10/11 00:00:00.00	Aurangabad	Y
4	4	PFname_4 PMname_4 PLname_4	Male	1995/06/04 00:00:00.00	Aurangabad	Y
5	5	PFname_5 PMname_5 PLname_5	Male	1995/09/24 00:00:00.00	Mumbai	Y
6	6	PFname_6 PMname_6 PLname_6	Male	1995/12/04 00:00:00.00	Mumbai	Y
7	7	PFname_7 PMname_7 PLname_7	Male	1995/10/01 00:00:00.00	Pune	Y
8	8	PFname_8 PMname_8 PLname_8	Female	1995/09/13 00:00:00.00	Mumbai	Y
9	9	PFname_9 PMname_9 PLname_9	Male	1995/05/25 00:00:00.00	Pune	Y
10	10	PFname_10 PMname_10 PLname_10	Male	1995/06/11 00:00:00.00	Nasik	Y
11	11	PFname_11 PMname_11 PLname_11	Female	1995/11/30 00:00:00.00	Pune	Y
12	12	PFname_12 PMname_12 PLname_12	Male	1995/08/02 00:00:00.00	Aurangabad	Y
13	13	PFname_13 PMname_13 PLname_13	Female	1995/11/16 00:00:00.00	Aurangabad	Y
14	14	PFname_14 PMname_14 PLname_14	Female	1995/07/31 00:00:00.00	Pune	Y
15	15	PFname_15 PMname_15 PLname_15	Female	1995/12/24 00:00:00.00	Aurangabad	Y
16	16	PFname_16 PMname_16 PLname_16	Male	1995/08/08 00:00:00.00	Pune	Y
17	17	PFname_17 PMname_17 PLname_17	Female	1995/07/12 00:00:00.00	Mumbai	Y
18	18	PFname_18 PMname_18 PLname_18	Male	1995/05/29 00:00:00.00	Nasik	Y
19	19	PFname_19 PMname_19 PLname_19	Male	1995/07/16 00:00:00.00	Pune	Y
20	20	PFname_20 PMname_20 PLname_20	Male	1995/07/05 00:00:00.00	Nasik	Y
21	21	PFname_21 PMname_21 PLname_21	Male	1995/07/27 00:00:00.00	Pune	Y
22	22	PFname_22 PMname_22 PLname_22	Female	1995/12/08 00:00:00.00	Pune	Y
23	23	PFname_23 PMname_23 PLname_23	Female	1995/08/14 00:00:00.00	Pune	Y
24	24	PFname_24 PMname_24 PLname_24	Male	1995/10/13 00:00:00.00	Nagpur	Y
25	25	PFname_25 PMname_25 PLname_25	Male	1995/05/21 00:00:00.00	Aurangabad	Y

c. Step Select Values

Melakukan pemilihan kolom yang relevan dan penyesuaian metadata (seperti tipe data atau nama kolom).



d. Step Insert/Update (Load)

Memasukkan data ke tabel dim_patient di database tujuan. Step ini juga berfungsi untuk memperbarui data pasien jika ada perubahan (*Update*).



e. Eksekusi Transformasi (Run)

Menjalankan transformasi HIMS_dim patient.ktr.

Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input	0	0	54049	54049	0	0	0	0	Finished	5.4s	10,044	-
2	Select values	0	54049	54049	0	0	0	0	0	Finished	6.8s	7,998	-
3	Insert / update	0	54049	54049	54049	54049	0	0	0	Finished	8.1s	6,685	-

f. Hasil di MySQL (Cek)

Memverifikasi data pada tabel dim_patient di database hims_dw untuk memastikan nama dan atribut lain tersimpan dengan benar.

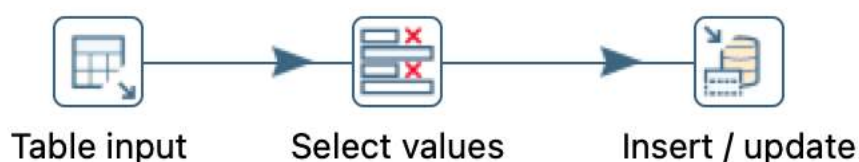
	patient_key	source_patient_id	full_name	gender_name	date_of_birth	city_name	status
<input type="checkbox"/>	Edit Copy Delete	1	1 Pfname_1 PMname_1 PLname_1	Male	1995-08-29	Aurangabad	Y
<input type="checkbox"/>	Edit Copy Delete	2	2 Pfname_2 PMname_2 PLname_2	Female	1995-08-20	Mumbai	Y
<input type="checkbox"/>	Edit Copy Delete	3	3 Pfname_3 PMname_3 PLname_3	Male	1995-10-11	Aurangabad	Y
<input type="checkbox"/>	Edit Copy Delete	4	4 Pfname_4 PMname_4 PLname_4	Male	1995-06-04	Aurangabad	Y
<input type="checkbox"/>	Edit Copy Delete	5	5 Pfname_5 PMname_5 PLname_5	Male	1995-09-24	Mumbai	Y
<input type="checkbox"/>	Edit Copy Delete	6	6 Pfname_6 PMname_6 PLname_6	Male	1995-12-04	Mumbai	Y
<input type="checkbox"/>	Edit Copy Delete	7	7 Pfname_7 PMname_7 PLname_7	Male	1995-10-01	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	8	8 Pfname_8 PMname_8 PLname_8	Female	1995-09-13	Mumbai	Y
<input type="checkbox"/>	Edit Copy Delete	9	9 Pfname_9 PMname_9 PLname_9	Male	1995-05-25	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	10	10 Pfname_10 PMname_10 PLname_10	Male	1995-06-11	Nasik	Y
<input type="checkbox"/>	Edit Copy Delete	11	11 Pfname_11 PMname_11 PLname_11	Female	1995-11-30	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	12	12 Pfname_12 PMname_12 PLname_12	Male	1995-08-02	Aurangabad	Y
<input type="checkbox"/>	Edit Copy Delete	13	13 Pfname_13 PMname_13 PLname_13	Female	1995-11-16	Aurangabad	Y
<input type="checkbox"/>	Edit Copy Delete	14	14 Pfname_14 PMname_14 PLname_14	Female	1995-07-31	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	15	15 Pfname_15 PMname_15 PLname_15	Female	1995-12-24	Aurangabad	Y
<input type="checkbox"/>	Edit Copy Delete	16	16 Pfname_16 PMname_16 PLname_16	Male	1995-08-08	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	17	17 Pfname_17 PMname_17 PLname_17	Female	1995-07-12	Mumbai	Y
<input type="checkbox"/>	Edit Copy Delete	18	18 Pfname_18 PMname_18 PLname_18	Male	1995-05-29	Nasik	Y
<input type="checkbox"/>	Edit Copy Delete	19	19 Pfname_19 PMname_19 PLname_19	Male	1995-07-16	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	20	20 Pfname_20 PMname_20 PLname_20	Male	1995-07-05	Nasik	Y
<input type="checkbox"/>	Edit Copy Delete	21	21 Pfname_21 PMname_21 PLname_21	Male	1995-07-27	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	22	22 Pfname_22 PMname_22 PLname_22	Female	1995-12-08	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	23	23 Pfname_23 PMname_23 PLname_23	Female	1995-08-14	Pune	Y
<input type="checkbox"/>	Edit Copy Delete	24	24 Pfname_24 PMname_24 PLname_24	Male	1995-10-13	Nagpur	Y
<input type="checkbox"/>	Edit Copy Delete	25	25 Pfname_25 PMname_25 PLname_25	Male	1995-05-21	Aurangabad	Y

4.5 Proses ETL: Load Tabel Dimensi Dokter (dim_doctor)

Transformasi data dokter melibatkan penggabungan nama dan pengambilan spesialisasi departemen.

a. Desain Transformasi (HIMS_dim doctor)

Membuat desain transformasi untuk memproses data dokter dari sumber ke target.



b. Step Table Input (Extract)

Mengambil data dari tabel doctors di database OLTP, termasuk ID dan nama-nama dokter.

Table input

Step name: Table input
Connection: hims

SQL

```
SELECT
d.DoctorID AS source_doctor_id,
CONCAT_WS(' ', d.FirstName, d.MiddleName, d.LastName) AS full_name,
dep.Name AS department_name,
g.Name AS gender_name,
d.Status AS status
FROM doctor d
LEFT JOIN department dep ON dep.DepartmentID = d.DepartmentID
LEFT JOIN gender g ON g.GenderID = d.GenderID;
```

Line 1 Column 0

☐ Store column info in step meta data
☐ Enable lazy conversion
☐ Replace variables in script?
☐ Insert data from step
☐ Execute for each row?
Limit size: 0

Help OK Preview Cancel

Rows of step: Table input (60 rows)

i	source_doctor_id	full_name	department_name	gender_name	status
1	1	Dr.DFname-1-1 DMname-1-1 DLname-1-1	Cardiology	Male	Y
2	2	Dr.DFname-1-2 DMname-1-2 DLname-1-2	Cardiology	Male	Y
3	3	Dr.DFname-1-3 DMname-1-3 DLname-1-3	Cardiology	Female	Y
4	4	Dr.DFname-1-4 DMname-1-4 DLname-1-4	Cardiology	Female	Y
5	5	Dr.DFname-1-5 DMname-1-5 DLname-1-5	Cardiology	Female	Y
6	6	Dr.DFname-2-1 DMname-2-1 DLname-2-1	Gynaecology	Male	Y
7	7	Dr.DFname-2-2 DMname-2-2 DLname-2-2	Gynaecology	Female	Y
8	8	Dr.DFname-2-3 DMname-2-3 DLname-2-3	Gynaecology	Female	Y
9	9	Dr.DFname-2-4 DMname-2-4 DLname-2-4	Gynaecology	Male	Y
10	10	Dr.DFname-2-5 DMname-2-5 DLname-2-5	Gynaecology	Male	Y
11	11	Dr.DFname-3-1 DMname-3-1 DLname-3-1	Medicine	Female	Y
12	12	Dr.DFname-3-2 DMname-3-2 DLname-3-2	Medicine	Female	Y
13	13	Dr.DFname-3-3 DMname-3-3 DLname-3-3	Medicine	Male	Y
14	14	Dr.DFname-3-4 DMname-3-4 DLname-3-4	Medicine	Female	Y
15	15	Dr.DFname-3-5 DMname-3-5 DLname-3-5	Medicine	Female	Y
16	16	Dr.DFname-4-1 DMname-4-1 DLname-4-1	Nephrology	Male	Y
17	17	Dr.DFname-4-2 DMname-4-2 DLname-4-2	Nephrology	Male	Y
18	18	Dr.DFname-4-3 DMname-4-3 DLname-4-3	Nephrology	Female	Y
19	19	Dr.DFname-4-4 DMname-4-4 DLname-4-4	Nephrology	Female	Y
20	20	Dr.DFname-4-5 DMname-4-5 DLname-4-5	Nephrology	Female	Y
21	21	Dr.DFname-5-1 DMname-5-1 DLname-5-1	Ophthalmology	Female	Y
22	22	Dr.DFname-5-2 DMname-5-2 DLname-5-2	Ophthalmology	Male	Y
23	23	Dr.DFname-5-3 DMname-5-3 DLname-5-3	Ophthalmology	Female	Y
24	24	Dr.DFname-5-4 DMname-5-4 DLname-5-4	Ophthalmology	Female	Y
25	25	Dr.DFname-5-5 DMname-5-5 DLname-5-5	Ophthalmology	Female	Y

c. Step Select Values

Memilih kolom yang diperlukan dan mengubah metadata, termasuk penggabungan nama depan, tengah, dan belakang jika diperlukan.

Step name:

Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to Nor
1	source_doctor_id		Integer	15	0	N
2	full_name		String	452		N
3	department_name		String	50		N
4	gender_name		String	50		N
5	status		String			N

Get fields to change

Help OK Cancel

d. Step Insert/Update (Load)

Memasukkan data dokter ke tabel dim_doctor di database tujuan. Step ini memastikan data dokter tersinkronisasi dengan baik.

Step name:

Connection: Edit... New... Wizard...

Target schema: Browse...

Target table: Browse...

Commit size:

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	source_doctor_id	=	source_doctor_id	

Get fields

Update fields:

#	Table field	Stream field	Update
1	source_doctor_id	source_doctor_id	N
2	full_name	full_name	Y
3	department_name	department_name	Y
4	gender_name	gender_name	Y
5	status	status	Y

Get update fields

Edit mapping

Help OK Cancel SQL

e. Eksekusi Transformasi (Run)

Menjalankan transformasi HIMS_dim doctor.ktr dan memantau log eksekusi.

Execution Results												
Logging Execution History Step Metrics Performance Graph Metrics Preview data												
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s) input/output
1	Table input	0	0	60	60	0	0	0	0	Finished	0.0s	6,667 -
2	Select values	0	60	60	0	0	0	0	0	Finished	0.0s	5,454 -
3	Insert / update	0	60	60	60	60	0	0	0	Finished	0.0s	2,400 -

f. Hasil di MySQL (Cek)

Memverifikasi tabel dim_doctor pada database hims_dw untuk memastikan semua dokter telah terdaftar.

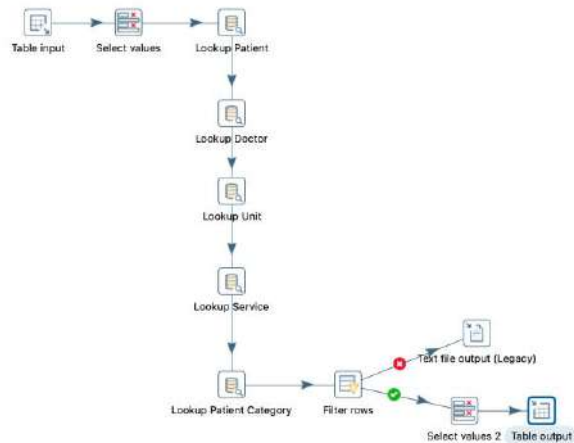
	doctor_key	source_doctor_id	full_name	department_name	gender_name	status
<input type="checkbox"/>	Edit	Copy	Delete	1	1 Dr.DFname-1-1 DMname-1-1 DLname-1-1	Cardiology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	2	2 Dr.DFname-1-2 DMname-1-2 DLname-1-2	Cardiology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	3	3 Dr.DFname-1-3 DMname-1-3 DLname-1-3	Cardiology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	4	4 Dr.DFname-1-4 DMname-1-4 DLname-1-4	Cardiology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	5	5 Dr.DFname-1-5 DMname-1-5 DLname-1-5	Cardiology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	6	6 Dr.DFname-2-1 DMname-2-1 DLname-2-1	Gynaecology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	7	7 Dr.DFname-2-2 DMname-2-2 DLname-2-2	Gynaecology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	8	8 Dr.DFname-2-3 DMname-2-3 DLname-2-3	Gynaecology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	9	9 Dr.DFname-2-4 DMname-2-4 DLname-2-4	Gynaecology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	10	10 Dr.DFname-2-5 DMname-2-5 DLname-2-5	Gynaecology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	11	11 Dr.DFname-3-1 DMname-3-1 DLname-3-1	Medicine Female Y
<input type="checkbox"/>	Edit	Copy	Delete	12	12 Dr.DFname-3-2 DMname-3-2 DLname-3-2	Medicine Female Y
<input type="checkbox"/>	Edit	Copy	Delete	13	13 Dr.DFname-3-3 DMname-3-3 DLname-3-3	Medicine Male Y
<input type="checkbox"/>	Edit	Copy	Delete	14	14 Dr.DFname-3-4 DMname-3-4 DLname-3-4	Medicine Female Y
<input type="checkbox"/>	Edit	Copy	Delete	15	15 Dr.DFname-3-5 DMname-3-5 DLname-3-5	Medicine Female Y
<input type="checkbox"/>	Edit	Copy	Delete	16	16 Dr.DFname-4-1 DMname-4-1 DLname-4-1	Nephrology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	17	17 Dr.DFname-4-2 DMname-4-2 DLname-4-2	Nephrology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	18	18 Dr.DFname-4-3 DMname-4-3 DLname-4-3	Nephrology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	19	19 Dr.DFname-4-4 DMname-4-4 DLname-4-4	Nephrology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	20	20 Dr.DFname-4-5 DMname-4-5 DLname-4-5	Nephrology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	21	21 Dr.DFname-5-1 DMname-5-1 DLname-5-1	Ophthalmology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	22	22 Dr.DFname-5-2 DMname-5-2 DLname-5-2	Ophthalmology Male Y
<input type="checkbox"/>	Edit	Copy	Delete	23	23 Dr.DFname-5-3 DMname-5-3 DLname-5-3	Ophthalmology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	24	24 Dr.DFname-5-4 DMname-5-4 DLname-5-4	Ophthalmology Female Y
<input type="checkbox"/>	Edit	Copy	Delete	25	25 Dr.DFname-5-5 DMname-5-5 DLname-5-5	Ophthalmology Female Y

4.6 Proses ETL: Load Tabel Fakta (fact_charge)

Proses ini adalah inti dari Data Warehouse, di mana data transaksi dari hospital_charges dihubungkan dengan seluruh tabel dimensi. Alur transformasi ini lebih kompleks karena melibatkan banyak *lookup* dan validasi.

a. Desain Transformasi (HIMS_fact charge)

Menampilkan alur lengkap transformasi yang dimulai dari ekstraksi data transaksi, pencarian kunci dimensi (*lookup*), penyaringan data (*filter*), hingga pemisahan antara data yang valid dan data yang gagal diproses.



b. Step Table Input (Extract)

Mengambil data transaksi dari tabel `hospital_charges` pada database sumber (OLTP). Data ini mencakup nilai transaksi (*amount*), diskon (*concession*), dan berbagai ID referensi (ID pasien, dokter, dll).

Table input

Step name

Table input

Connection

hims

Edit...

New...

Wizard...

SQL

Get SQL select statement...

```

SELECT
  ch.ChargeId AS source_charge_id,

  -- tentukan encounter
  CASE
    WHEN ch.AdmissionId IS NOT NULL THEN 'IPD'
    ELSE 'OPD'
  END AS encounter_type,

  ch.VisitId AS source_visit_id,
  ch.AdmissionId AS source_admission_id,

  -- date_key ambil dari VisitDate/AdmissionDate, fallback ke AddedDateTime
  CAST(
    DATE_FORMAT(
      COALESCE(v.VisitDate, a.AdmissionDate, ch.AddedDateTime),
      '%Y%m%d'
    ) AS UNSIGNED
  ) AS date_key,

  -- natural keys untuk lookup ke dim
  COALESCE(v.PatientId, a.PatientId) AS source_patient_id,
  COALESCE(v.DoctorID, a.DoctorId) AS source_doctor_id,
  COALESCE(v.UnitId, a.UnitId) AS source_unit_id,
  COALESCE(v.PatientCategoryID, a.PatientCategoryID) AS source_category_id,
  ch.ServiceId AS source_service_id,

  -- measures
  ch.Rate AS rate,
  ch.Quantity AS quantity,
  ch.Amount AS gross_amount,
  IFNULL(ch.Concession, 0) AS concession_amount,
  (ch.Amount - IFNULL(ch.Concession, 0)) AS net_amount

FROM charge ch
LEFT JOIN visit v ON v.VisitId = ch.VisitId
LEFT JOIN admission a ON a.AdmissionId = ch.AdmissionId;

```

Line 1 Column 0

Store column info in step meta data

Enable lazy conversion

Replace variables in script?

Insert data from step

Execute for each row?

Limit size 0

Help

OK

Preview

Cancel

Serangkaian step *Database Lookup* digunakan untuk mengganti ID operasional (seperti doctor_id) dengan *Surrogate Key* (seperti doctor_key) dari tabel dimensi di Data Warehouse. Urutan lookup yang dilakukan:

1. **Lookup Patient:** Mencari patient_key.

The screenshot shows the 'Database lookup' dialog box with the following configuration:

- Step name: **Lookup Patient**
- Connection: **hims_dw**
- Lookup schema: **hims_dw**
- Lookup table: **dim_patient**
- Enable cache?: ☒
- Cache size in rows (0=cache everything): **0**
- Load all data from table: ☐
- The key(s) to look up the value(s):

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

#	Field	New name	Default	Type
1	patient_key			None
- Do not pass the row if the lookup fails: ☐
- Fail on multiple results?: ☐
- Order by:

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

2. **Lookup Doctor:** Mencari doctor_key.

The screenshot shows the 'Database lookup' dialog box with the following configuration:

- Step name: **Lookup Doctor**
- Connection: **hims_dw**
- Lookup schema: **hims_dw**
- Lookup table: **dim_doctor**
- Enable cache?: ☒
- Cache size in rows (0=cache everything): **0**
- Load all data from table: ☐
- The key(s) to look up the value(s):

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

#	Field	New name	Default	Type
1	doctor_key			None
- Do not pass the row if the lookup fails: ☐
- Fail on multiple results?: ☐
- Order by:

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

3. Lookup Unit: Mencari unit_key.

The screenshot shows the 'Database lookup' dialog box with the following configuration:

- Step name: **Lookup Unit**
- Connection: **hims_dw**
- Lookup schema: **hims_dw**
- Lookup table: **dim_unit**
- Enable cache?: ☒
- Cache size in rows (0=cache everything): **0**
- Load all data from table: ☐
- The key(s) to look up the value(s):

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

#	Field	New name	Default	Type
1	unit_key		None	
- Do not pass the row if the lookup fails: ☐
- Fail on multiple results?: ☐
- Order by:
- Buttons: **Help**, **OK**, **Cancel**, **Get Fields**, **Get lookup fields**

4. Lookup Service: Mencari service_key.

The screenshot shows the 'Database lookup' dialog box with the following configuration:

- Step name: **Lookup Service**
- Connection: **hims_dw**
- Lookup schema: **hims_dw**
- Lookup table: **dim_service**
- Enable cache?: ☒
- Cache size in rows (0=cache everything): **0**
- Load all data from table: ☐
- The key(s) to look up the value(s):

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

#	Field	New name	Default	Type
1	service_key		None	
- Do not pass the row if the lookup fails: ☐
- Fail on multiple results?: ☐
- Order by:
- Buttons: **Help**, **OK**, **Cancel**, **Get Fields**, **Get lookup fields**

5. Lookup Patient Category: Mencari patient_category_key.

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

- Step name:** Lookup Patient Category
- Connection:** hims_dw
- Lookup schema:** hims_dw
- Lookup table:** dim_patient_category
- Enable cache?** ☒
- Cache size in rows (0=cache everything):** 0
- Load all data from table:** ☐
- The key(s) to look up the value(s):**

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

#	Field	New name	Default	Type
1	patient_category_key			None
- Do not pass the row if the lookup fails:** ☐
- Fail on multiple results?** ☐
- Order by:**

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

e. Step Filter Rows (Validasi Data)

Validasi dilakukan untuk memastikan semua data transaksi memiliki referensi yang valid ke dimensi. Step ini mengecek apakah kunci dimensi (misal patient_key) bernilai NULL atau tidak ditemukan.

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

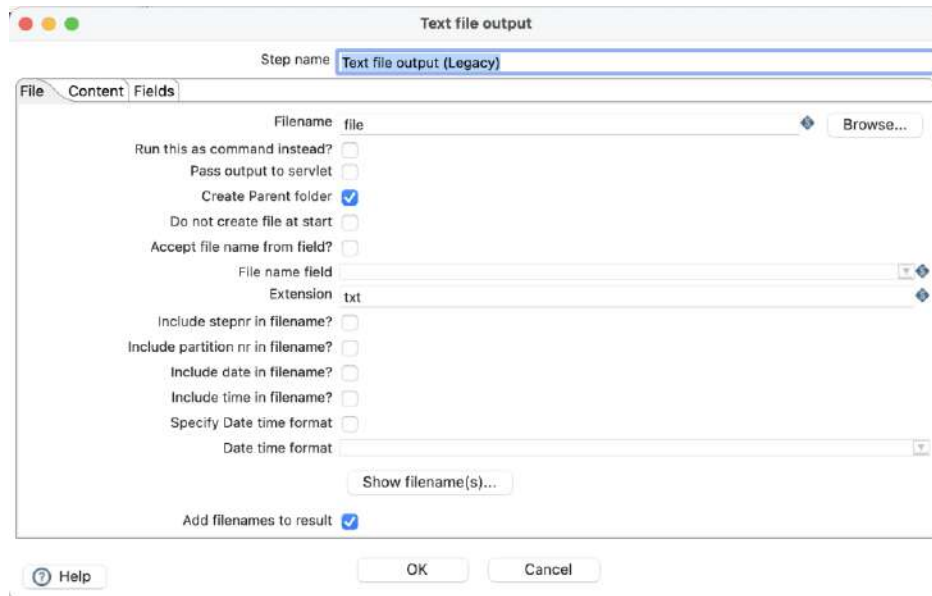
- Step name:** Filter rows
- Send 'true' data to step:** Select values 2
- Send 'false' data to step:** Text file output (Legacy)
- The condition:**

```
patient_key IS NOT NULL
AND
doctor_key IS NOT NULL
AND
unit_key IS NOT NULL
AND
service_key IS NOT NULL
```

Buttons at the bottom: Help, OK, Cancel.

f. Alur Data Salah

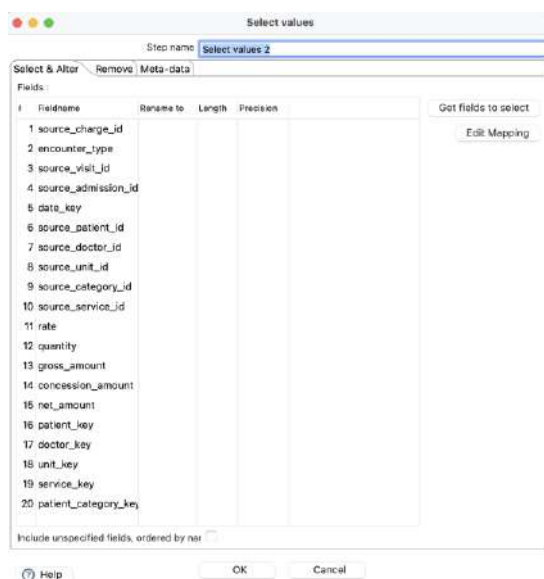
Text File Output Jika kondisi Filter bernilai False (data tidak valid/kunci tidak ditemukan), baris data tersebut dibuang ke dalam file teks log error (error_log.txt) agar tidak mengotori Data Warehouse dan bisa diinvestigasi kemudian.



g. Alur Data Benar (True Path) Jika data lolos validasi (kondisi True), alur berlanjut ke tahap finalisasi dan pemuatan:

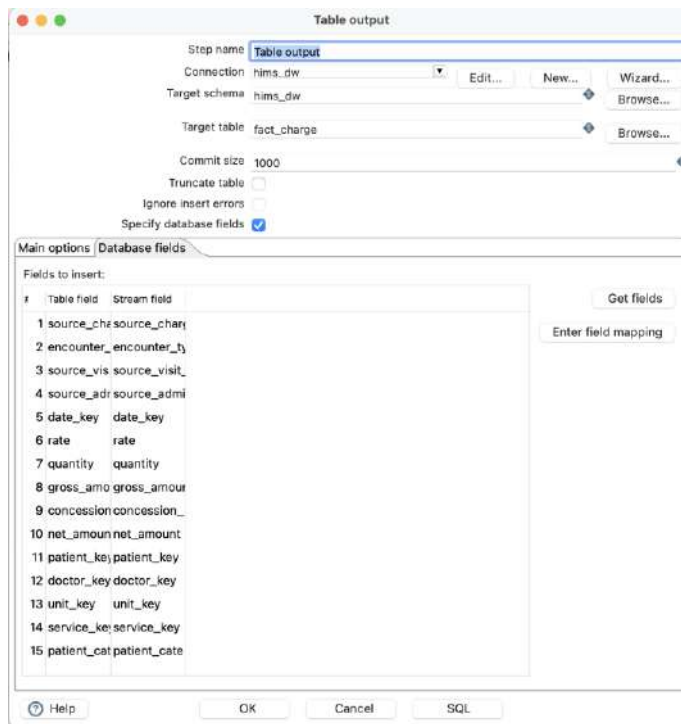
1. Select Values (Finalisasi)

Pada tahap ini (sesuai gambar teknis *Select values 2*), dilakukan pemilihan field final yang akan masuk ke gudang data. Kolom yang dipilih meliputi *source ID* (seperti *source_charge_id*, *source_visit_id*) untuk jejak audit, *Surrogate Keys* hasil lookup (*patient_key*, *doctor_key*, *date_key*, dll), serta metrik nilai (*gross_amount*, *net_amount*). Metadata juga dipastikan presisinya.



2. Table Output (Load)

Langkah terakhir mengirimkan data ke tabel fact_charge di koneksi hims_dw. Opsi *Specify database fields* diaktifkan untuk memetakan secara manual kolom dari *Stream* (hasil olahan Pentaho) ke *Table Field* (kolom fisik di database), memastikan data seperti net_amount masuk ke kolom yang tepat.



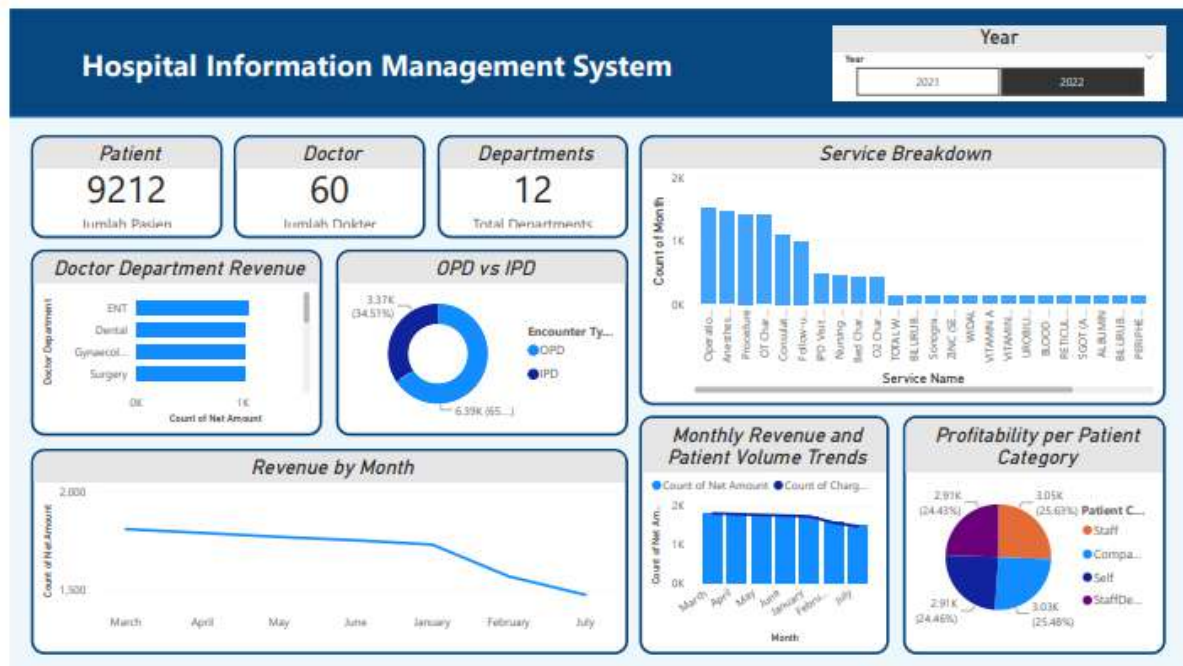
BAB 5

HASIL DAN PEMBAHASAN

5.1 Implementasi Dashboard Business Intelligence

Setelah proses ETL berhasil memindahkan dan mentransformasi data dari sistem operasional (OLTP) ke dalam Data Warehouse (OLAP), tahap selanjutnya adalah visualisasi data. Dashboard ini dibangun untuk memberikan gambaran menyeluruh mengenai kinerja rumah sakit, mencakup aspek volume pasien, kinerja dokter, dan tren pendapatan.

Dashboard ini terhubung langsung dengan *view* vw_charge_enriched yang telah disiapkan sebelumnya, memungkinkan manajemen untuk melihat metrik kunci tanpa perlu menjalankan kueri SQL manual.



5.2 Analisis Key Performance Indicators (KPI)

Pada bagian atas dashboard, terdapat ringkasan metrik utama (*High-level Metrics*) yang menunjukkan skala operasional rumah sakit saat ini.

Total Volume Pasien Tercatat sebanyak **9.212** pasien unik yang telah dilayani dalam periode data yang tersedia. Angka ini menunjukkan basis data pasien yang cukup besar yang dikelola oleh rumah sakit.

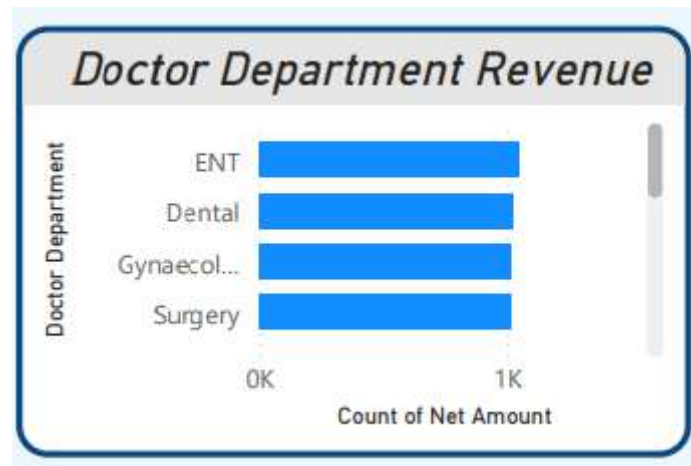
Sumber Daya Medis Rumah sakit memiliki **60** dokter yang aktif memberikan layanan. Rasio ini dapat digunakan manajemen untuk mengevaluasi apakah jumlah tenaga medis sudah memadai untuk menangani volume pasien yang ada.

Cakupan Unit Layanan Layanan terdistribusi ke dalam **12** departemen atau unit spesialisasi yang berbeda.

5.3 Analisis Dimensi Operasional

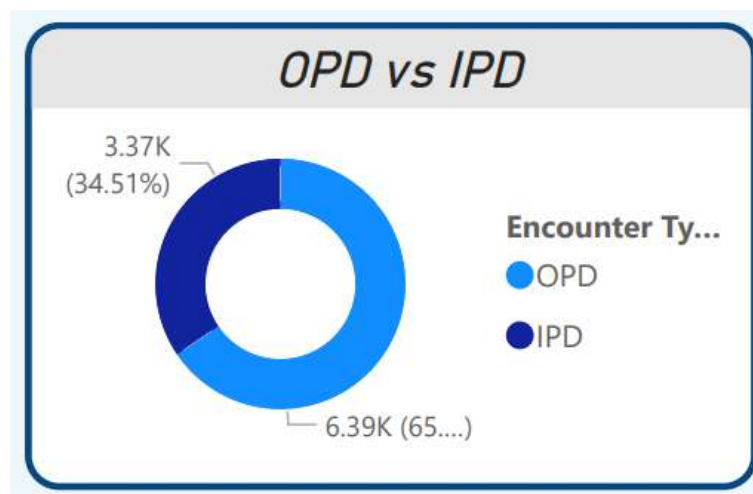
Bagian ini membedah data berdasarkan dimensi-dimensi yang telah dirancang dalam skema bintang (*Star Schema*).

A. Analisis Departemen dan Dokter (Doctor Department Revenue) Visualisasi batang (*bar chart*) menampilkan kontribusi pendapatan atau volume transaksi berdasarkan departemen. Departemen seperti THT (*ENT*), Gigi (*Dental*), Ginekologi (*Gynaecology*), dan Bedah (*Surgery*) terlihat memberikan kontribusi signifikan. Analisis ini membantu manajemen mengidentifikasi unit mana yang menjadi "pusat profit" (*profit center*) utama rumah sakit.



B. Perbandingan Jenis Kunjungan (OPD vs IPD)

Diagram lingkaran (*pie chart*) menunjukkan proporsi antara layanan Rawat Jalan (*Out-Patient Department/OPD*) dan Rawat Inap (*In-Patient Department/IPD*). Berdasarkan visualisasi, terlihat dominasi satu jenis layanan sebesar **65,49% (6,39 Ribu)** dibandingkan layanan lainnya yang sebesar **34,51% (3,37 Ribu)**. Hal ini mengindikasikan bahwa mayoritas volume transaksi rumah sakit berasal dari salah satu sektor tersebut (asumsi: umumnya OPD memiliki volume lebih tinggi, namun IPD memiliki nilai per transaksi lebih besar).

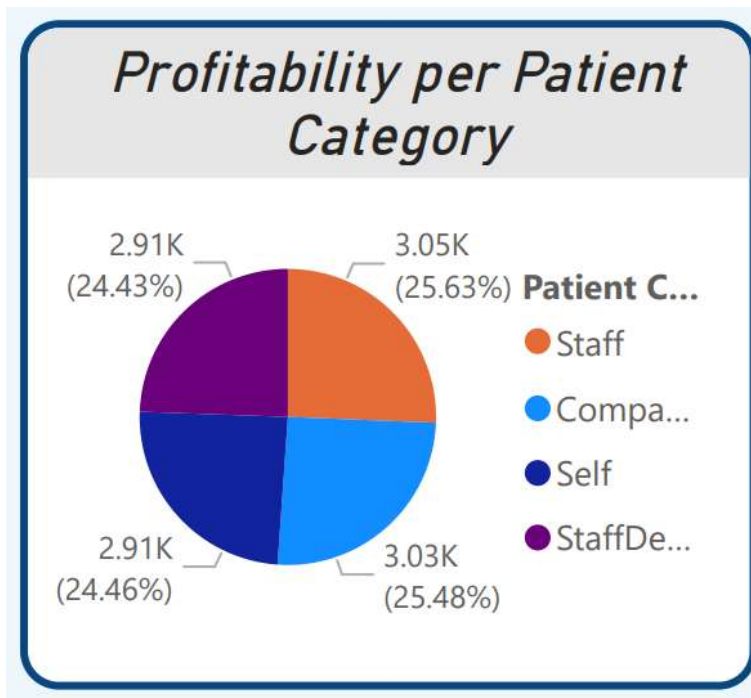


C. Profitabilitas Kategori Pasien

Salah satu temuan menarik dari dashboard ini adalah distribusi kategori penjamin bayaran pasien. Diagram *Profitabilitas per Patient Category* menunjukkan pembagian yang sangat merata, masing-masing berkisar antara **24% hingga 25%**, untuk kategori:

- Staff
- Company (Asuransi/Perusahaan)
- Self (Bayar Mandiri)
- Staff Dependent (Tanggungans Staf)

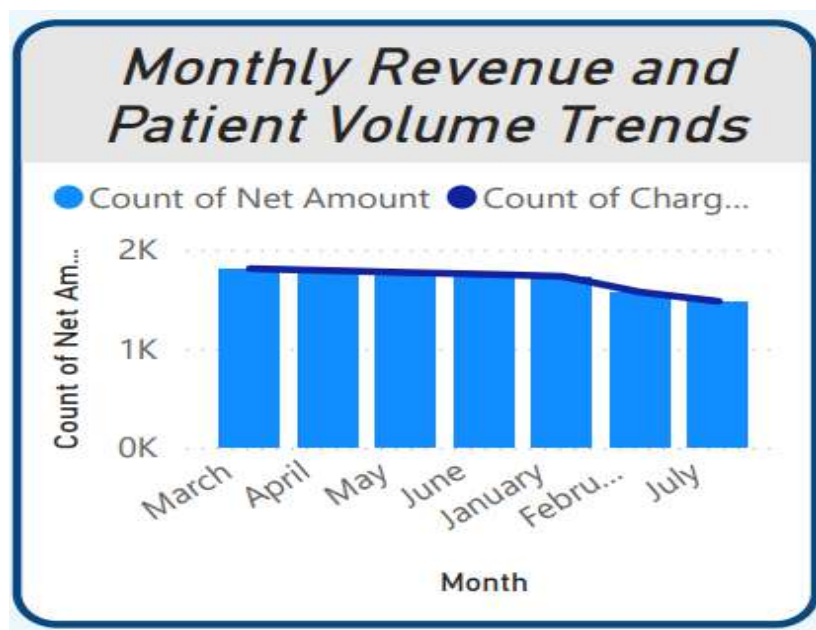
Keseimbangan ini menunjukkan bahwa rumah sakit tidak memiliki ketergantungan risiko pada satu jenis pembayar saja.



5.4 Analisis Tren Waktu (Time Series Analysis)

Dimensi waktu (*dim_date*) memungkinkan analisis tren bulanan yang krusial untuk melihat pola musiman.

Tren Pendapatan Bulanan Grafik *Revenue by Month* dan *Monthly Revenue Trends* memperlihatkan fluktuasi kinerja dari bulan Januari hingga Juli. Terlihat adanya variasi volume transaksi pada bulan-bulan tertentu (misalnya Maret, April, Mei). Informasi ini berguna bagi manajemen untuk merencanakan alokasi staf atau logistik obat-obatan pada bulan-bulan sibuk.



5.5 Temuan Penting (Key Findings)

Berdasarkan keseluruhan analisis dashboard, dapat disimpulkan beberapa temuan bisnis sebagai berikut:

1. Stabilitas Kategori Pasien Distribusi pasien antara kategori mandiri, perusahaan, dan staf sangat seimbang. Ini menandakan kebijakan diversifikasi pasien rumah sakit berjalan baik.
2. Kinerja Departemen Terdapat variasi beban kerja antar 12 departemen yang ada. Departemen bedah dan spesialis tertentu menunjukkan aktivitas tinggi yang berkorelasi dengan pendapatan.
3. Pola Kunjungan Dengan total 9.212 pasien yang ditangani oleh 60 dokter, rata-rata beban kerja per dokter perlu diperhatikan untuk menjaga kualitas layanan, terutama pada bulan-bulan puncak kunjungan.

BAB 6

PENUTUP

6.1 Kesimpulan

Berdasarkan seluruh rangkaian kegiatan implementasi Business Intelligence pada Hospital Information Management System (HIMS), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Keberhasilan Integrasi Data Proyek ini sukses mengintegrasikan data operasional yang terfragmentasi (pasien, dokter, transaksi) ke dalam satu pusat data (*Data Warehouse*) yang terstruktur menggunakan skema bintang (*Star Schema*). Hal ini menjawab masalah kesulitan akses data lintas modul yang sebelumnya dihadapi.
2. Efektivitas ETL Proses *Extract, Transform, Load* (ETL) menggunakan Pentaho Data Integration terbukti efektif dalam membersihkan data mentah, menstandarisasi format (seperti penggabungan nama dan format tanggal), serta menghitung metrik bisnis (*net_amount*) secara otomatis.
3. Peningkatan Kemampuan Analisis Dengan terbentuknya tabel fakta *fact_charge* dan dimensi pendukung, manajemen kini dapat melakukan analisis multidimensi. Dashboard yang dihasilkan mampu memvisualisasikan total 9.212 pasien dan kinerja 60 dokter secara *at-a-glance*, menggantikan proses pelaporan manual yang lambat.
4. Wawasan Bisnis yang Jelas Sistem BI ini berhasil mengungkap pola-pola penting, seperti kontribusi pendapatan per departemen (contoh: dominasi bedah/THT) dan keseimbangan proporsi kategori penjamin bayaran, yang sebelumnya sulit terlihat pada data transaksi mentah.

6.2 Saran

Untuk pengembangan sistem Business Intelligence HIMS yang lebih lanjut dan komprehensif, penulis menyarankan beberapa hal:

1. Automasi Terjadwal (Scheduling) Mengimplementasikan penjadwalan otomatis (*Job Scheduling*) pada Pentaho agar proses pembaruan data dari OLTP ke Data Warehouse berjalan setiap malam tanpa intervensi manual.
2. Perluasan Modul Menambahkan subjek area lain ke dalam Data Warehouse, seperti manajemen inventaris farmasi dan logistik, untuk mendapatkan gambaran efisiensi biaya operasional yang lebih lengkap.
3. Visualisasi Interaktif Mengembangkan dashboard yang lebih interaktif menggunakan alat seperti Power BI atau Tableau yang terkoneksi langsung ke database `hims_dw`, memungkinkan pengguna untuk melakukan *drill-down* data hingga ke level detail transaksi harian.
4. Analisis Prediktif Memanfaatkan data historis yang sudah tersimpan rapi untuk melakukan analisis prediktif, seperti meramalkan lonjakan kunjungan pasien di bulan-bulan tertentu untuk optimalisasi jadwal jaga dokter.