

Analisis Penggunaan Kamar Pasien dan Jumlah Penyakit Diderita di Rumah Sakit Hasan Sadikin Jawa Barat

Patricia Joanne¹, Shofiyyah Nadhiroh²

^{1,2}Departemen Ilmu Komputer, Universitas Padjadjaran, Indonesia

Article Info

Article history:

Received May 23, 2019

Revised Jun 5, 2019

Accepted Jun 20, 2019

Keywords:

Data warehouse

OLAP

Rumah sakit

SQL

ABSTRACT (10 PT)

Banyaknya data yang dimiliki oleh pihak rumah sakit haruslah dikelola dengan baik karena jika data yang banyak ini belum terkelola dengan baik akan mengakibatkan pimpinan rumah sakit mengalami keterbatasan dalam mengambil keputusan dengan cepat dan akhirnya memiliki kesulitan dalam melihat kinerja dan mutu rumah sakit yang dipimpinnya. Dalam penelitian ini akan dirancang sebuah data warehouse untuk mengelola data yang berfokus kepada bagaimana pimpinan melihat kinerja dan mutu rumah sakit yang dipimpinnya dari sisi penggunaan kamar dalam rumah sakit yang digunakan oleh pasien dan jumlah diagnosis terbanyak yang diderita oleh pasien. Diharapkan dengan adanya data warehouse, maka mutu pelayanan rumah sakit dapat ditingkatkan dan dapat disesuaikan dengan standar mutu nasional yang ada.

Copyright © 2019 Departemen Ilmu Komputer.
All rights reserved.

Corresponding Author:

Patricia Joanne,
Departemen Ilmu Komputer,
Universitas Padjadjaran,
Jl. Raya Bandung Sumedang KM. 21, Hegarmanah, Jatinangor, Kabupaten Sumedang, Jawa Barat 45363
Email: patricia16002@mail.unpad.ac.id

1. PENDAHULUAN

Kebutuhan masyarakat akan rumah sakit tidak bisa dielakkan lagi, mulai dari proses kelahiran, pengobatan untuk anak maupun dewasa, keadaan gawat darurat yang membutuhkan rawat inap maupun rawat jalan sampai pemeriksaan rutin yang membutuhkan bantuan dari laboratorium misalnya radiologi, ronsen, dan lain-lain. Hal ini akan mengakibatkan banyaknya data yang dimiliki oleh rumah sakit. Data tersebut berguna untuk memberikan informasi dengan cepat dan akurat kepada pihak internal yaitu mulai dari pimpinan rumah sakit sampai pihak operasional yang diberi wewenang dan pihak eksternal yaitu pasien sendiri.

Banyaknya data yang dimiliki oleh pihak rumah sakit haruslah dikelola dengan baik karena jika data yang banyak ini belum terkelola dengan baik akan mengakibatkan pimpinan rumah sakit mengalami keterbatasan dalam mengambil keputusan dengan cepat dan akhirnya memiliki kesulitan dalam melihat kinerja dan mutu rumah sakit yang dipimpinnya. Dalam penelitian ini akan dirancang sebuah *data warehouse* untuk mengelola data yang berfokus kepada bagaimana pimpinan melihat kinerja dan mutu rumah sakit yang dipimpinnya dari sisi penggunaan kamar dalam rumah sakit yang digunakan oleh pasien dan jumlah diagnosis terbanyak yang diderita oleh pasien.

Berdasarkan hal tersebut maka dapat dirangkum tujuan perancangan *data warehouse* ini adalah agar pimpinan rumah sakit dapat menganalisis tren jenis kamar pasien rawat inap selama satu tahun dan dapat menganalisis jumlah diagnosis penyakit terbanyak yang diderita oleh pasien rawat inap dan rawat jalan selama satu tahun.

2. STUDI LITERATUR

2.1. Definisi Data Warehouse

Terdapat beberapa definisi *data warehouse* dari berbagai ahli, diantaranya adalah sebagai berikut.

Menurut W.H. Inmon, *data warehouse* adalah kumpulan data yang memiliki sifat *subject-oriented*, *integrated*, *timeline*, dan *non-volatile* dalam pengumpulan data untuk mendukung proses pengambilan keputusan manajemen.

Menurut Ralph Kimball, *data warehouse* adalah suatu sistem yang mengekstrak, melakukan pembersihan, menyesuaikan diri serta mengirimkan sumber data pada suatu data penyimpanan dimensional dan selanjutnya memberi dukungan pada implementasi query serta analisa dalam rangka pengambilan keputusan.

Dengan demikian, data warehouse merupakan metode dalam perancangan *database*, yang menunjang DSS (*decision support system*) dan EIS (*executive information system*). Secara fisik *data warehouse* adalah database, tetapi perancangan data warehouse dan database sangat berbeda. Desain database tradisional menggunakan normalisasi, sedangkan pada data warehouse normalisasi bukanlah cara terbaik.

2.2. Karakteristik Data Warehouse

Berikut ini adalah karakteristik dari *data warehouse*:

1. Berorientasi kepada subjek (*Subject-oriented*)

Data Warehouse berorientasi subjek artinya *data warehouse* didesain untuk menganalisis data berdasarkan subjek-subjek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu. *Data warehouse* diorganisasikan di sekitar subjek-subjek utama dari perusahaan (*customers*, *products* dan *sales*) dan tidak diorganisasikan pada area-area aplikasi utama (*customer invoicing*, *stock control* dan *product sales*). Hal ini dikarenakan kebutuhan dari data warehouse untuk menyimpan data-data yang bersifat sebagai penunjang suatu keputusan, dari pada aplikasi yang berorientasi terhadap data. Jadi dengan kata lain, data yang disimpan adalah berorientasi kepada subjek bukan terhadap proses.

2. Data yang dimiliki terintegrasi (*Data Integrated*)

Data Warehouse dapat menyimpan data-data yang berasal dari sumber-sumber yang terpisah ke dalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Dengan demikian data tidak bisa dipecah-pecah karena data yang ada merupakan suatu kesatuan yang menunjang keseluruhan konsep data warehouse itu sendiri.

Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara seperti konsisten dalam penamaan variabel, konsisten dalam ukuran variabel, konsisten dalam struktur pengkodean dan konsisten dalam atribut fisik dari data.

3. Dibuat dalam rentang waktu tertentu (*Timeline*)

Seluruh data pada *data warehouse* dapat dikatakan akurat atau valid pada rentang waktu tertentu.

4. Data yang disimpan bersifat tetap (*Non-volatile*)

Karakteristik keempat dari data warehouse adalah *non-volatile*, maksudnya data pada *data warehouse* tidak di-update secara *real time* tetapi di *refresh* dari sistem operasional secara reguler. Data yang baru selalu ditambahkan sebagai suplemen bagi database itu sendiri dari pada sebagai sebuah perubahan. Database tersebut secara kontinu menyerap data baru ini, kemudian secara inkremental disatukan dengan data sebelumnya.

2.3. Komponen Data Warehouse

Untuk membuat sebuah *data warehouse*, dibutuhkan komponen-komponen yaitu fakta dan dimensi.

2.3.1. Tabel fakta

Menurut Connolly dan Begg (2005, p1183), tabel fakta adalah “*every dimensional model (DM) is composed of one table with a composite primary key, called the fact table*,” yang berarti tabel fakta adalah satu tabel pada model dimensional yang isinya *composite primary key*.

2.3.2. Tabel dimensi

Menurut Connolly dan Begg (2005, p1183), tabel dimensi adalah “*a set of smaller tables called dimension tables*,” yang berarti tabel dimensi adalah sekumpulan tabel-tabel yang lebih kecil dari tabel fakta pada model dimensional. Setiap tabel dimensi mempunyai *non-composite primary key*.

2.4. Metodologi Data Warehouse

Ada beberapa metodologi yang dapat membantu dalam perancangan *data warehouse*, salah satunya adalah metodologi yang dikemukakan oleh Kimball.

Berdasarkan kutipan dalam Conolly dan Begg, metodologi yang dikemukakan oleh Kimball dalam membangun data warehouse ada sembilan tahapan.

1. Pemilihan proses
2. Pemilihan sumber
3. Mengidentifikasi dimensi
4. Pemilihan fakta
5. Menyimpan pre-kalkulasi di tabel fakta
6. Melengkapi tabel dimensi
7. Pemilihan durasi database
8. Menelusuri perubahan dimensi yang perlahan
9. Menentukan prioritas dan mode *query*

Pada tahun 2002, Kimball menyederhanakan tahapan tersebut menjadi empat tahap. Empat tahap tersebut adalah sebagai berikut:

1. Memilih proses bisnis (*Select the business process*).
2. Menyatakan granularity proses bisnis (*Declare the grain*).
3. Menentukan dimensi untuk setiap baris tabel fakta (*Identify the dimensions*).
4. Mengidentifikasi fakta numerik yang akan mengisi setiap baris tabel fakta (*Identify the facts*).

2.5. Skema Data Warehouse

Dalam perancangan *data warehouse*, dibutuhkan skema untuk dapat menggambarkan tabel fakta dan tabel dimensinya. Ada 3 jenis skema *data warehouse* yaitu *snowflake*, *star*, dan *fact constellation*. Dalam perancangan *data warehouse* ini kami menggunakan skema *fact constellation*.

Skema *Fact Constellation* memiliki beberapa tabel fakta yang menggunakan satu atau beberapa table dimensi secara bersama-sama sehingga jika digambarkan akan terlihat seperti sekumpulan bintang. Skema ini banyak digunakan dan lebih kompleks daripada skema bintang dan skema *snowflake*. Skema *fact constellation* juga dikenal skema *galaxy*.

2.6. ETL (Extract, Transform, Load)

ETL adalah proses dalam pergudangan data yang bertanggung jawab untuk mengambil data dari sumber dan menempatkannya ke dalam gudang data. ETL melibatkan tugas-tugas berikut:

- Extract – mengekstraksi data dari sumber (SAP, ERP, dan sistem operasional lainnya), data dari berbagai sumber diubah menjadi satu format datawarehouse dan siap untuk pemrosesan transformasi
- Transform – proses transformasi data dapat melibatkan beberapa tugas, seperti derivasi, pembersihan data, filter data, pemecahan kolom, penggabungan data, transposisi baris dan kolom, dan lain-lain.
- Load – memuat data ke gudang data.

2.7. OLAP (Online Analytical Processing)

OLAP adalah teknologi di balik aplikasi Business Intelligence (BI). OLAP adalah teknologi canggih untuk penemuan data, termasuk kemampuan untuk melihat laporan tanpa batas, perhitungan analitik yang rumit, dan perencanaan skenario “bagaimana jika” (anggaran, perkiraan).

3. HASIL DAN ANALISIS

3.1. Kebutuhan Penelitian

Berikut ini adalah kebutuhan yang harus dipersiapkan dalam penelitian ini baik spesifikasi hardware dan software yang dibutuhkan.

1. Spesifikasi *hardware*

Spesifikasi yang tertulis di bawah ini adalah spesifikasi dari PC yang akan digunakan dalam penelitian, dalam hal ini spesifikasi berikut bukan berarti spesifikasi minimum.

- OS : Windows 10
- RAM : 8 GB
- HDD : 1 TB

2. *Software* dan *file* yang dibutuhkan

Berikut ini adalah software yang dibutuhkan dalam perancangan *data warehouse*.

- Microsoft SQL Server 2017
- Dataset RS Hasan Sadikin Jawa Barat (*dummy*)

3.2. Tahapan Penelitian

Metode yang digunakan adalah metode Kimball (2002) yang memiliki 4 tahapan. Metode ini adalah metode Kimball yang telah disederhanakan dari 9 tahapan menjadi 4 tahapan.

3.2.1. Memilih proses bisnis

Pemilihan proses bisnis dilakukan untuk memperjelas batasan data warehouse yang akan dibuat.

Adapun proses yang dipilih adalah:

- Registrasi rawat inap
- Registrasi rawat jalan
- Rekam medis pasien

3.2.2. Menyatakan granularity proses bisnis

Grain merupakan calon fakta yang dapat dianalisis. Pemilihan grain dilakukan untuk memutuskan apa yang direpresentasikan record dari tabel fakta. Grain yang digunakan dalam perancangan data warehouse ini yaitu:

- Registrasi rawat inap → Fact_kamarpasien
- Rekam medis pasien → Fact_diagpenyakit

3.2.3. Menentukan dimensi untuk setiap baris tabel fakta

Berikut ini identifikasi dimensi untuk setiap tabel fakta:

Dimensi Grain	Kamar Pasien	Diagnosis Penyakit Pasien
Waktu	v	v
Dokter	v	v
Pasien	v	v
Diagnosis	v	v
Transaksi	v	v
Kamar	v	v

3.2.4. Mengidentifikasi fakta numerik yang akan mengisi setiap baris tabel fakta

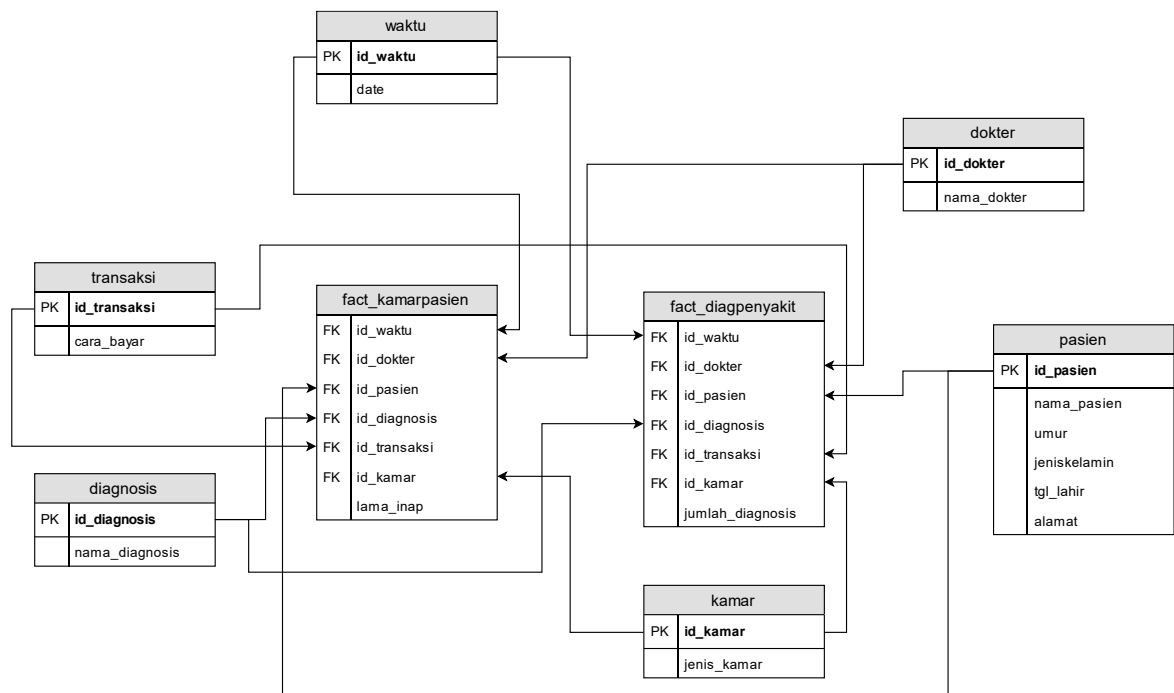
Sesuai dengan grain yang telah ditentukan sebelumnya, masing-masing fakta memiliki data yang dapat dihitung untuk ditampilkan dalam bentuk tabel atau grafik:

- Registrasi rawat inap / kamar pasien:
 - Jumlah pasien rawat inap mendaftar
 - Jumlah kamar pasien digunakan berdasarkan jenisnya
 - Rata-rata lama penggunaan kamar pasien
- Rekam medis pasien / diagnosis penyakit pasien
 - Jumlah pasien berdasarkan diagnosisnya

3.4. Hasil Penelitian

3.4.1. Skema

Skema yang digunakan dalam penelitian ini adalah skema *fact constellation*.



3.4.2. Hasil ETL

Kami memiliki dataset berisi 100 baris data *dummy* untuk menunjang perancangan *data warehouse* ini. Dataset ini terdiri dari kolom-kolom yaitu id_waktu, tgl, bln, thn, id_dokter, nama_dokter, id_kamar, jenis_kamar, id_pasien, nama_pasien, umur, jenis_kelamin, tgl_lahir, alamat, id_transaksi, cara_bayar, id_diagnosis, dan nama_diagnosis.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	id_waktu	tgl	bln	thn	id_dokter	nama_dokter	id_kamar	jenis_kamar	id_pasien	nama_pasien	umur	jenis_kelamin	tgl_lahir	alamat	id_transaksi	cara_bayar	id_diagnosis	nama_diagnosis	
2	20170303	03	03	2017	2003	Luthfi Seti	4002	SW	3002	Chelsea M	19	Perempuan	17-08-98	Jl Wastuki	7003	Debit	6005	Septisemia	
3	20190801	08	01	2019	2023	Cawuk Wi	4006	VIP	3003	Balian Pui	22	Laki-laki	27-07-97	Jl Yohanes	7003	Debit	6009	Stroke	
4	20183003	30	03	2018	2003	Luthfi Seti	4007	VVIP	3004	Kunthara	32	Laki-laki	17-07-86	Jl Hegarmi	7006	AXA Manc	6002	Diare dan muntaber	
5	20192501	25	01	2019	2012	Ikhsan Ma	4003	III	3005	Artawan R	53	Laki-laki	09-06-66	Jl Samoja	7005	BPJS	6009	Stroke	
6	20180202	02	02	2018	2007	Estiawan I	4002	SW	3006	Zalindra P	25	Perempuan	24-03-93	Jl Jend A Y	7005	BPJS	6007	Anemia	
7	20181402	14	02	2018	2008	Kawaca Sil	4002	SW	3007	Karma Ha	44	Perempuan	08-02-74	Jl Hegarmi	7005	BPJS	6009	Stroke	
8	20191101	11	01	2019	2024	Asirwandi	4003	III	3008	Eli Wulanc	60	Perempuan	08-06-59	Jl Hegarmi	7003	Debit	6004	Pneumonia	
9	20172312	23	12	2017	2006	Tirta Hida	4003	III	3009	Vicky Mus	27	Laki-laki	23-07-90	Jl Wastuki	7005	BPJS	6003	Gagal jantung	
10	20180710	07	10	2018	2010	Vivi Hidayi	4003	III	3010	Labuh Yuli	56	Perempuan	05-11-62	Jl Tubagus	7003	Debit	6009	Stroke	
11	20193004	30	04	2019	2018	Jasmani N	4010	Intermedi	3011	Oliva Meli	28	Perempuan	05-10-91	Jl Terusan	7004	Kredit	6005	Septisemia	
12	20170712	07	12	2017	2006	Tirta Hida	4006	VIP	3012	Putri Sudic	43	Perempuan	05-01-74	Jl Hegarmi	7004	Kredit	6003	Gagal jantung	
13	20182201	22	01	2018	2010	Vivi Hidayi	4009	HCU-RIK	3013	Nyoman I	29	Perempuan	25-11-89	Jl Griya Ra	7002	Tunai	6009	Stroke	
14	20171309	13	09	2017	2008	Kawaca Sil	4010	Intermedi	3014	Martana Z	57	Perempuan	08-05-60	Jl Jend A Y	7006	AXA Manc	6002	Diare dan muntaber	
15	20171908	19	08	2017	2021	Aris Palast	4010	Intermedi	3015	Damu Hal	37	Perempuan	20-06-80	Jl Wastuki	7004	Kredit	6003	Gagal jantung	
16	20173110	31	10	2017	2008	Kawaca Sil	4006	VIP	3016	Jelita Marj	17	Perempuan	01-05-00	Jl Kopo S1	7002	Tunai	6006	Gagal ginjal	
17	20180401	04	01	2018	2022	Manah Gu	4006	VIP	3017	Kania Siml	61	Perempuan	29-04-57	Jl Terusan	7003	Debit	6003	Gagal jantung	
18	20170102	01	02	2017	2012	Ikhsan Ma	4004	II	3018	Kemal Nat	29	Laki-laki	30-05-88	Jl Wastuki	7002	Tunai	6007	Anemia	
19	20191502	15	02	2019	2019	Tania Ram	4011	Intensive t	3019	Cinthia Wi	32	Perempuan	31-12-87	Jl Samoja	7004	Kredit	6003	Gagal jantung	
20	20180701	07	01	2018	2014	Winda Na	4009	HCU-RIK	3020	Gaduh Ha	45	Laki-laki	19-02-73	Jl Babakar	7004	Kredit	6005	Septisemia	
21	20172608	26	08	2017	2008	Kawaca Sil	4002	SW	3021	Jasmin An	40	Perempuan	29-10-77	Jl Wastuki	7005	BPJS	6006	Gagal ginjal	

Dataset tersebut harus melalui proses ETL agar dapat digunakan dalam *data warehouse*.

1. Extract

Extract adalah proses memilih dan mengambil data dari satu atau beberapa sumber dan membaca data yang dipilih tersebut. Menggunakan SQL Server, berikut ini contoh mengambil data diagnosis penyakit dari dataset.

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [id_diagnosis]
, [nama_diagnosis]
FROM [rshs].[dbo].[dimen_diagnosis]

```

	id_diagnosis	nama_diagnosis
1	6002	Diare dan muntaber
2	6003	Gagal jantung
3	6004	Pneumonia
4	6005	Septisemia
5	6006	Gagal ginjal
6	6007	Anemia
7	6008	Tuberkulosis (TBC)
8	6009	Stroke

2. Transform

Transform adalah proses membersihkan dan mengubah data dari bentuk asli menjadi bentuk yang sesuai dengan kebutuhan data warehouse. Menggunakan SQL Server, dalam kasus ini data pasien yang telah diextract diubah tipe datanya.

Import Flat File 'rshs'

Modify Columns

Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results

Modify Columns
This operation generated the following table schema. Please verify if schema is accurate, and if not, please make any changes.

Column Name	Data Type	Primary Key	Allow Nulls
id_pasien	int	<input type="checkbox"/>	<input type="checkbox"/>
nama_pasien	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
umur	int	<input type="checkbox"/>	<input type="checkbox"/>
jeniskelamin	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
tgl_lahir	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
alamat	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>

Import Flat File 'rshs'

Modify Columns

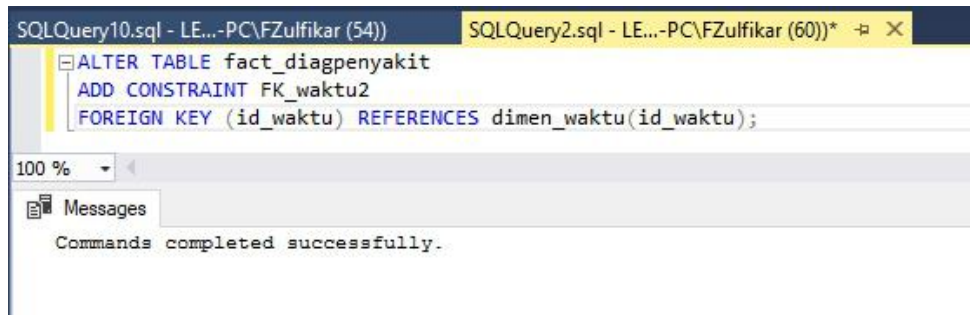
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results

Modify Columns
This operation generated the following table schema. Please verify if schema is accurate, and if not, please make any changes.

Column Name	Data Type	Primary Key	Allow Nulls
id_pasien	varchar(50)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
nama_pasien	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
umur	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
jeniskelamin	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
tgl_lahir	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
alamat	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>

3. Load

Load adalah proses memasukkan data ke dalam data warehouse. Dalam kasus ini, foreign key waktu ditambahkan pada tabel fakta diagpenyakit.



```
SQLQuery10.sql - LE...-PC\FZulfikar (54)) SQLQuery2.sql - LE...-PC\FZulfikar (60))*
ALTER TABLE fact_diagpenyakit
ADD CONSTRAINT FK_waktu2
FOREIGN KEY (id_waktu) REFERENCES dimen_waktu(id_waktu);

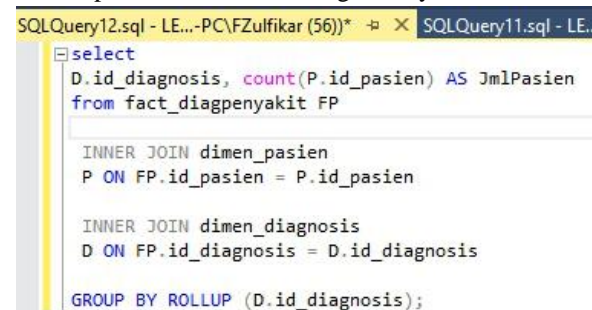
100 %
Messages
Commands completed successfully.
```

3.4.3. Hasil OLAP

Berikut ini adalah hasil pengolahan OLAP. Terdapat 5 jenis *query* yang mendukung OLAP yang akan dicontohkan sebagai berikut.

1. ROLLUP

Jumlah pasien berdasarkan diagnosisnya dan total seluruh diagnosis.

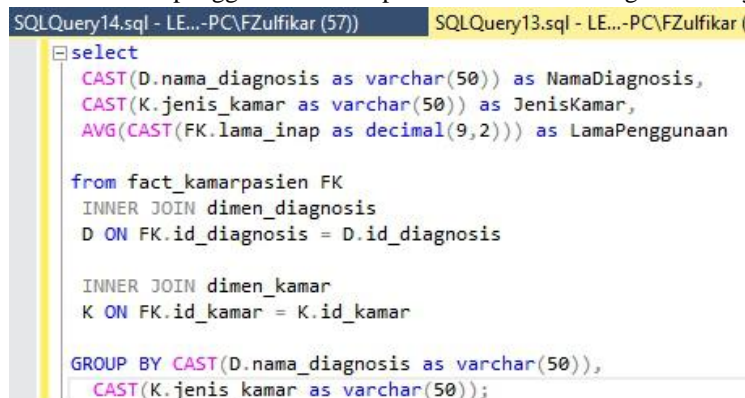


```
SQLQuery12.sql - LE...-PC\FZulfikar (56))* SQLQuery11.sql - LE...
select
D.id_diagnosis, count(P.id_pasien) AS JmlPasien
from fact_diagpenyakit FP
INNER JOIN dimen_pasien
P ON FP.id_pasien = P.id_pasien
INNER JOIN dimen_diagnosis
D ON FP.id_diagnosis = D.id_diagnosis
GROUP BY ROLLUP (D.id_diagnosis);
```

	id_diagnosis	JmlPasien
1	6002	15
2	6003	15
3	6004	6
4	6005	12
5	6006	14
6	6007	14
7	6008	9
8	6009	15
9	NULL	100

2. DRILLDOWN

Rata-rata lama penggunaan kamar pasien berdasarkan diagnosis dan jenis kamarnya



```
SQLQuery14.sql - LE...-PC\FZulfikar (57)) SQLQuery13.sql - LE...-PC\FZulfikar (
select
CAST(D.nama_diagnosis as varchar(50)) as NamaDiagnosis,
CAST(K.jenis_kamar as varchar(50)) as JenisKamar,
AVG(CAST(FK.lama_inap as decimal(9,2))) as LamaPenggunaan
from fact_kamarpasien FK
INNER JOIN dimen_diagnosis
D ON FK.id_diagnosis = D.id_diagnosis
INNER JOIN dimen_kamar
K ON FK.id_kamar = K.id_kamar
GROUP BY CAST(D.nama_diagnosis as varchar(50)),
CAST(K.jenis_kamar as varchar(50));
```


Results		Messages	
	NamaDiagnosis	JenisKamar	LamaPenggunaan
1	Diare dan muntaber	HCU-RIK	12.000000
2	Gagal jantung	HCU-RIK	11.666666
3	Septisemia	HCU-RIK	15.000000
4	Stroke	HCU-RIK	18.500000
5	Anemia	I	12.000000
6	Diare dan muntaber	I	8.000000
7	Gagal ginjal	I	6.500000
8	Pneumonia	I	3.000000
9	Septisemia	I	16.500000
10	Tuberkulosis (TBC)	I	7.000000
11	Anemia	II	20.000000
12	Diare dan muntaber	II	20.000000
13	Gagal ginjal	II	18.000000
14	Pneumonia	II	10.000000
15	Septisemia	II	8.000000
16	Stroke	II	8.000000
17	Tuberkulosis (TBC)	II	17.000000
18	Anemia	III	18.000000
19	Diare dan muntaber	III	11.000000
20	Gagal jantung	III	10.000000
21	Pneumonia	III	11.500000
22	Septisemia	III	2.000000
23	Stroke	III	6.000000
24	Anemia	Intensive ...	15.000000
25	Gagal ginjal	Intensive ...	3.000000
26	Gagal jantung	Intensive ...	9.500000
27	Septisemia	Intensive ...	14.000000
28	Stroke	Intensive ...	17.000000
29	Anemia	Intemedi...	18.000000
30	Diare dan muntaber	Intemedi...	7.800000
31	Gagal ginjal	Intemedi...	8.000000
32	Gagal jantung	Intemedi...	11.666666
33	Septisemia	Intemedi...	10.500000
34	Stroke	Intemedi...	14.500000
35	Tuberkulosis (TBC)	Intemedi...	5.000000
36	Anemia	Suite	7.000000
37	Gagal ginjal	Suite	14.000000
38	Gagal jantung	Suite	4.500000
39	Septisemia	Suite	16.000000
40	Stroke	Suite	17.000000
41	Tuberkulosis (TBC)	Suite	13.000000
42	Anemia	SW	11.000000
43	Diare dan muntaber	SW	12.500000
44	Gagal ginjal	SW	7.500000
45	Septisemia	SW	3.000000
46	Stroke	SW	10.500000
47	Tuberkulosis (TBC)	SW	10.000000
48	Anemia	VIP	16.000000
49	Diare dan muntaber	VIP	4.500000

50	Gagal ginjal	VIP	2.000000
51	Gagal jantung	VIP	13.333333
52	Pneumonia	VIP	19.000000
53	Stroke	VIP	15.000000
54	Diare dan muntaber	VVIP	3.000000
55	Gagal ginjal	VVIP	16.000000
56	Pneumonia	VVIP	8.000000
57	Septisemia	VVIP	12.000000
58	Stroke	VVIP	18.000000
59	Tuberkulosis (TBC)	VVIP	10.500000

3. SLICING

Waktu kedatangan, nama pasien dirawat dan jenis kamar yang digunakan oleh dokter Tania Ramadan

SQLQuery14.sql - LE...-PC\FZulfikar (57) SQLQuery13.sql

```

select
W.date,
P.nama_pasien,
K.jenis_kamar

from fact_diagpenyakit FP
INNER JOIN dimen_waktu
W ON FP.id_waktu = W.id_waktu
INNER JOIN dimen_dokter
D ON FP.id_dokter = D.id_dokter
INNER JOIN dimen_pasien
P ON FP.id_pasien = P.id_pasien
INNER JOIN dimen_kamar
K ON FP.id_kamar = K.id_kamar

where D.nama_dokter like 'Tania Ramadan'

```

Results Messages

	date	nama_pasien	jenis_kamar
1	2017-01-29	Viktor Hutapea	Intermediate Care-Isolasi
2	2017-07-07	Mursita Nugroho	Intensive Care
3	2018-03-14	Almira Fujiati	VVIP
4	2019-02-15	Cinthia Wahyudin	Intensive Care
5	2019-03-02	Tina Januar	Suite
6	2019-05-21	Ajeng Safitri	VVIP

4. DICING

Jumlah pasien berdasarkan diagnosisnya

SQLQuery14.sql - LE...-PC\FZulfikar (57) SQLQuery13.sql - LE...

```

select
D.id_diagnosis, count(P.id_pasien) AS JmlPasien
from fact_diagpenyakit FP

INNER JOIN dimen_pasien
P ON FP.id_pasien = P.id_pasien

INNER JOIN dimen_diagnosis
D ON FP.id_diagnosis = D.id_diagnosis

GROUP BY (D.id_diagnosis);

```

Results Messages		
	id_diagnosis	JmlPasien
1	6002	15
2	6003	15
3	6004	6
4	6005	12
5	6006	14
6	6007	14
7	6008	9
8	6009	15

5. CUBING

Cube jumlah pasien berdasarkan jenis kamar dan diagnosisnya

SQLQuery14.sql - LE...-PC\FZulfikar (57) SQLQuery13.sql - LE...

```

select
    K.jenis_kamar,
    D.nama_diagnosis,
    COUNT(P.id_pasien) AS JmlPasien

from fact_kamarpasien FK

    INNER JOIN dimen_kamar
    K ON FK.id_kamar = K.id_kamar
    INNER JOIN dimen_diagnosis
    D ON FK.id_diagnosis = D.id_diagnosis
    INNER JOIN dimen_pasien
    P ON FK.id_pasien = P.id_pasien

GROUP BY cube (D.nama_diagnosis,K.jenis_kamar)

ORDER BY (K.jenis_kamar)

```

Results Messages			
	jenis_kamar	nama_diagnosis	JmlPasien
1	NULL	NULL	100
2	NULL	Anemia	14
3	NULL	Diare dan muntaber	15
4	NULL	Gagal ginjal	14
5	NULL	Gagal jantung	15
6	NULL	Pneumonia	6
7	NULL	Septisemia	12
8	NULL	Stroke	15
9	NULL	Tuberkulosis (TBC)	9
10	HCU-RIK	Diare dan muntaber	1
11	HCU-RIK	Gagal jantung	3
12	HCU-RIK	Septisemia	2

13	HCU-RIK	Stroke	2
14	HCU-RIK	NULL	8
15	I	Anemia	2
16	I	Diare dan muntaber	2
17	I	Gagal ginjal	2
18	I	Pneumonia	1
19	I	Septisemia	2
20	I	Tuberkulosis (TBC)	1
21	I	NULL	10
22	II	Anemia	1
23	II	Diare dan muntaber	1
24	II	Gagal ginjal	1
25	II	Pneumonia	1
26	II	Septisemia	1
27	II	Stroke	2
28	II	Tuberkulosis (TBC)	1
29	II	NULL	8
30	III	Anemia	1
31	III	Diare dan muntaber	1
32	III	Gagal jantung	2
33	III	Pneumonia	2
34	III	Septisemia	1
35	III	Stroke	2
36	III	NULL	9
37	Intensive ...	Anemia	2
38	Intensive ...	Gagal ginjal	1
39	Intensive ...	Gagal jantung	2
40	Intensive ...	Septisemia	1
41	Intensive ...	Stroke	1
42	Intensive ...	NULL	7
43	Intermedia...	Anemia	2
44	Intermedia...	Diare dan muntaber	5
45	Intermedia...	Gagal ginjal	2
46	Intermedia...	Gagal jantung	3
47	Intermedia...	Septisemia	2
48	Intermedia...	Stroke	2
49	Intermedia...	Tuberkulosis (TBC)	1
50	Intermedia...	NULL	17
51	Suite	Anemia	2
52	Suite	Gagal ginjal	2
53	Suite	Gagal jantung	2
54	Suite	Septisemia	1
55	Suite	Stroke	1
56	Suite	Tuberkulosis (TBC)	2
57	Suite	NULL	10
58	SW	Anemia	3
59	SW	Diare dan muntaber	2
60	SW	Gagal ginjal	4

61	SW	Septisemia	1
62	SW	Stroke	2
63	SW	Tuberkulosis (TBC)	2
64	SW	NULL	14
65	VIP	Anemia	1
66	VIP	Diare dan muntaber	2
67	VIP	Gagal ginjal	1
68	VIP	Gagal jantung	3
69	VIP	Pneumonia	1
70	VIP	Stroke	1
71	VIP	NULL	9
72	VVIP	Diare dan muntaber	1
73	VVIP	Gagal ginjal	1
74	VVIP	Pneumonia	1
75	VVIP	Septisemia	1
76	VVIP	Stroke	2
77	VVIP	Tuberkulosis (TBC)	2
78	VVIP	NULL	8

4. KESIMPULAN



Dengan akan diimplementasikannya aplikasi data warehouse pada rumah sakit, diharapkan dapat memenuhi kebutuhan dan keinginan informasi oleh pihak eksekutif rumah sakit yang terkait dengan kinerja rumah sakit.

Dengan kemampuan yang ditawarkan pada aplikasi ini, diharapkan ketersediaan informasi dari berbagai sudut pandang yang berbeda dapat memenuhi harapan dari pimpinan rumah sakit, yang kedepannya dapat digunakan untuk menyediakan layanan yang lebih baik dan menaikkan nilai bisnis.

REFERENSI

- Antonius, Henry dan Eka Widjaja. (2010). Data Warehouse pada Rumah Sakit. Jurnal. Fakultas Ilmu Komputer Universitas Bina Nusantara, Jakarta.
- Arwanto, Nandintyo. Pembuatan Data Warehouse Pengelolaan Perbekalan Farmasi Rumah Sakit XYZ. Jurnal. Institut Teknologi Sepuluh Nopember, Surabaya.
- Connolly, T.M. and Begg, C.E. (2005) Database Systems: A Practical Approach to Design, Implementation, and Management. 4th Edition, Pearson Education, Harlow.
- Inmon, W. (2005). Building The DataWarehouse fourth edition. Indiana: Wiley.
- Kimball, R. J. (2004). The Data Warehouse ETLToolkit. New Delhi: WILEY.
- Turban, E. (2005). Sistem Pendukung Keputusan dan Sistem Cerdas edisi 7 jilid 1. Yogyakarta: AndiOffset.

BIOGRAFI PENULIS

	<p>Patricia Joanne lahir pada tanggal 24 Juni 1999 dan berasal dari Jakarta. Saat ini mengambil jurusan Teknik Informatika di Universitas Padjadjaran.</p>
	<p>Shofiyyah Nadhiroh lahir pada tanggal 28 November 1999 dan berasal dari Jakarta. Saat ini mengambil jurusan Teknik Informatika di Universitas Padjadjaran.</p>