

TRANSFORMASI ONTOLOGI KE MODEL DIMENSIONAL

Parmonangan R. Togatorop*¹, Christina Simanjuntak², Christine Nababan³, Genii Silitonga³

^{1,2,3}Institut Teknologi Del, Laguboti

Email: ¹mona.togatorop@del.ac.id, ²christinalarasati27@gmail.com,

³christinenababan1213@gmail.com, ⁴geniiruben98@gmail.com

*Penulis Korespondensi

(Naskah masuk: 27 Oktober 2021, diterima untuk diterbitkan: 31 Januari 2022)

Abstrak

Data warehouse adalah salah satu komponen penting untuk analisis bisnis yang efektif. Salah satu model pada *data warehouse* adalah *dimensional model* yang banyak digunakan karena performa pemrosesan yang lebih cepat dari model lain. Dua faktor utama pada perancangan *dimensional model* adalah sumber data dan kebutuhan bisnis. Salah satu sumber data yang banyak digunakan adalah ontologi karena mampu merepresentasikan data menjadi informasi yang koheren yang dapat dimasukkan ke dalam Data Warehouse. Dalam penelitian ini dihasilkan sebuah tools berbasis ontologi yang digunakan untuk secara otomatis mendapatkan *dimensional model* untuk *data warehouse* dari sumber data dan kebutuhan bisnis. Tahapan yang dilakukan mengidentifikasi semua informasi pada file *ontology* yang dimasukkan ke dalam daftar dimensi dan *fact* dihasilkan berdasarkan aturan perancangan *dimensional model*. Pembuatan *fact* tabel dan dimensi tabel dirancang berdasarkan rule perancangan *dimensional model* yang diperkenalkan oleh Kimball. Setelah tabel *fact* dan dimensi diidentifikasi, maka tabel *fact* dan dimensi tersebut diubah ke dalam bentuk kueri yang dapat dieksekusi pada MySQL. Penelitian ini berhasil menghasilkan *dimensional model* dari sumber data *ontology*.

Kata kunci: *data warehouse, dimensional model, ontology*

A TOOL FOR ONTOLOGY TRANSFORMATION TO DIMENSIONAL MODEL

Abstract

The data warehouse is one of the essential components for effective business analysis. One of the models in the data warehouse is the dimensional model that is widely used because the processing performance is faster than other models. The two main factors in designing a dimensional model are data sources and business requirements. One of the most widely used data sources is ontology because it is able to represent data into coherent information that can be entered into the Data Warehouse. In this research, an ontology-based tool is produced which is used to automatically obtain dimensional models for the data warehouse from data sources and business needs. The step taken identifies all the information in the ontology file entered by the user, then a list of dimensions and facts is generated based on the design rules for the dimensional model. The creation of fact tables and dimension tables are designed based on the dimensional model design rules introduced by Kimball. After the fact and dimension tables are identified, the fact and dimension tables are converted into queries that can be executed in MySQL. This study succeeded in producing dimensional models from the ontology data source.

Keywords: *data warehouse, dimensional model, ontology*

1. PENDAHULUAN (huruf besar, 10pt, tebal)

Data pada basis data operasional tradisional tidak sepenuhnya dapat digunakan untuk pengambilan (Ren & Wang, 2018). Hal tersebut disebabkan oleh pertumbuhan data yang semakin meningkat seiring dengan waktu. Oleh karena pengoperasian data operasional dalam skala yang besar sulit dilakukan, teknologi *data warehouse* sering digunakan sebagai tempat penyimpanan data. *Database* operasional dan *data warehouse* perlu

dipisahkan karena karakteristik pemrosesan untuk lingkungan operasional dan *data warehouse* pada dasarnya berbeda (Scholtz, 2016). Data yang disimpan pada *data warehouse* akan dianalisis untuk proses pengambilan keputusan (Moulay & Drias, 2018). Pendekatan yang paling umum digunakan dalam perancangan *data warehouse* adalah *dimensional model* oleh Kimball (Scholtz, 2016). Implementasi *data warehouse* dapat membantu organisasi dalam menentukan strategi bisnis,

menyediakan informasi, meningkatkan proses bisnis dalam organisasi, memungkinkan pengaksesan data, dan mempertahankan keamanan data (G, et al., 2017). *Dimensional model* adalah salah satu model rancangan logical *data warehouse* dengan performa yang tinggi. Dimensional model oleh Kimball digunakan untuk memodelkan data mart karena lebih simpel dan kinerja kueri yang lebih baik (Yessad & Labiod, 2016).

Oleh karena itu, semakin baik model multidimensi yang diterapkan sesuai dengan kebutuhan analitis para pembuat keputusan, semakin bermanfaat analisis OLAP untuk proses pengambilan keputusan. Untuk mencapai tujuan tersebut, banyak upaya telah diinvestasikan oleh akademisi dan komunitas industri dalam merancang *Dimensional model* (Sautot, et al., 2015) [1]. Namun perancangan dan implementasi *data warehouse* membutuhkan pengetahuan dan level keahlian yang tinggi mengenai domain bisnis yang dikelola sehingga sering menyebabkan kegagalan dalam implementasi *data warehouse* (Yessad & Labiod, 2016). Beberapa hal yang harus dipertimbangkan dalam pembangunan *data warehouse* adalah desain *dimensional model* (Ren & Wang, 2018) dan sumber data (Moulay & Drias, 2018). Para *designer dimensional model* biasanya melakukan proses manual untuk menganalisis sumber data yang akan digunakan dalam pembuatan *dimensional model* dan selanjutnya membuat skema desain *dimensional model* secara manual dari sumber data yang telah dianalisis (Wamars & Randriatoamanana, 2016). Hal tersebut dilakukan karena tidak adanya model standar untuk mewakili struktur data warehouse (Bojičić, et al., 2016).

Beberapa penelitian memanfaatkan *ontology* karena dapat membantu dalam merepresentasikan data menjadi informasi yang sesuai ke dalam data warehouse, mampu merepresentasikan kebutuhan bisnis selama proses perancangan, dan memiliki kemampuan penalaran untuk memfasilitasi tugas perancangan. *Ontology* adalah salah satu solusi untuk standarisasi penamaan tabel dan *field* dalam basis data sehingga basis data yang dihasilkan akan memiliki penamaan yang sama dan dapat memudahkan pertukaran data (Rus & Othman, 2019). Teknologi web semantik seperti *ontology* dapat digunakan untuk membantu merepresentasikan data menjadi informasi yang koheren yang dapat dimasukkan ke dalam Data Warehouse (Thenmozhi, 2013).

Penelitian M. Thenmozhi dan K. Vivekanandan menghasilkan sebuah tool transformasi yang dikenal dengan OBDWSD (*Ontology Based Data Warehouse Schema Design*) (Thenmozhi, 2013). Penelitian Li juga menggunakan *ontology* untuk membangun tabel *fact dimensional model* (Li, et al., 2021). Selain itu, penelitian Sebaa, et al juga menggunakan *ontology* dalam merancang *dimensional model* dan mampu

melakukan transformasi *ontology* menjadi tabel dimensi dan *fact* (Sebaa, et al., 2018).

Thenmozhi mengusulkan OBDWSD (*Ontology Based Data Warehouse Schema Design*) untuk menghasilkan desain skema *dimensional model* berbasis *ontology* yang menggunakan metodologi hibrida (Thenmozhi, 2013). Pada OBDWSD, terdapat *fact* yang hanya berelasi pada satu dimensi serta komponen tabel *fact* dan dimensi yang dihasilkan belum lengkap karena belum memiliki *key* serta tidak memiliki tabel dimensi date. Hal ini tidak sesuai dengan pendekatan dimensional model menurut Kimball, dimana yang mengharuskan setiap tabel *fact* berelasi dengan minimal dua tabel dimensi, memiliki *key* yang menyatakan relasi antar tabel *fact* dan dimensi, serta tabel *fact* harus memiliki relasi dengan tabel dimensi date/ time. Oleh karena itu, penelitian ini akan mengadaptasikan proses pembuatan tabel *fact* dan dimensi untuk menghasilkan dimensional model penelitian OBDWSD agar sesuai dengan pendekatan Kimball dengan menerapkannya ke dalam sebuah tool untuk melihat perubahan yang dihasilkan baik sebelum atau setelah perbaikan proses pembuatan tabel *fact* dan dimensi.

Penelitian ini mengadaptasi proses pembuatan tabel *fact* dan dimensi untuk menghasilkan *dimensional model* menggunakan *ontology* dan menerapkannya ke dalam sebuah tool .

2. METODE PENELITIAN

2.1 Dimensional Model

Dimensional model adalah model atau teknik yang digunakan untuk menyajikan data analitik, dimana syaratnya adalah data yang dihasilkan dapat dipahami oleh pengguna bisnis dan menghasilkan performa kueri yang cepat. Konsep utama dalam *dimensional model* adalah *fact* , dimensi, dan atribut. Dimensi dapat memiliki hierarki yang berbeda serta memiliki atribut yang menjelaskan siapa, apa, dimana, dan mengapa dari suatu dimensional model. Grain atau tingkat granularitas adalah konsep yang menjelaskan tingkat kedetailan. *Fact*, dimensi, dan atribut dapat diorganisir dengan beberapa cara yang disebut skema. Pemilihan skema yang akan digunakan bergantung pada variabel keperluan apa yang diperlukan untuk pembangunan *data warehouse*, *Business Intelligence*, dan lainnya.

Berdasarkan pendekatan Kimball terdapat beberapa indikator yang menyebabkan komponen tersebut boleh atau tidak boleh terkandung dalam dimensional model. Pertama, *Business Process* dan *Granularity*. Proses bisnis adalah kegiatan operasional yang dilakukan oleh suatu organisasi/ perusahaan sedangkan *granularity* menetapkan representasi apa yang sebenarnya diwakili satu baris tunggal tabel *fact* . Analisis yang akan dilakukan terhadap *dimensional model* yang dihasilkan tidak akan mengevaluasi secara detail terhadap proses business process dan *granularity* yang dihasilkan

karena menggunakan data source *ontology*, dimana business process atau grain setiap *dimensional model* sudah berdasarkan data source yang tersedia.

Kedua adalah aturan terkait tabel Dimensi. Indikator suatu tabel dimensi menurut Kimball (Kimball & Ross, 2013) adalah memiliki surrogate *key* sebagai *primary key* dan tabel dimensi sederhana dan unique. Surrogate *key* menggunakan auto increment number dan akan dijadikan sebagai *primary key*. Selain itu *dimensional model* harus memiliki dimensi date/ time yang berelasi dengan *fact* (Feki & Hachaichi, 2013). Selain itu, tabel dimensi memiliki natural *key* sebagai *key* unik dalam database. Penelitian Mudasir (Kirmani, 2017) tidak memiliki natural *key*, namun menggunakan surrogate *key* sebagai *primary key*. Natural *key* tetapi tidak dijadikan sebagai *primary key* karena rentan mengalami perubahan (Aleksic, et al., 2010).

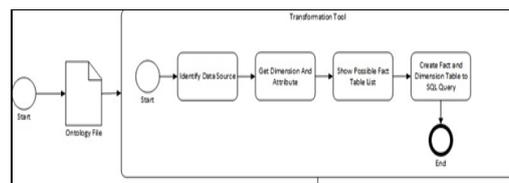
Aturan terkait Tabel Fact pada *dimensional model* adalah tabel *fact* memiliki atribut bertipe data numerik sebagai measure event (Kimball & Ross, 2013). Tabel *fact* memiliki *transaction*, *periodic*, dan *accumulating snapshot fact*. Selain itu, tabel *fact* juga harus memiliki foreign *key*. Tabel *fact* memiliki karakteristik berupa composite *key* yang tiap elemennya adalah foreign *key* yang diperoleh dari tabel dimensi. Tabel *fact* yang tidak memiliki atribut measure masih dapat dikategorikan sebagai *fact*, namun foreign *key* pada *fact* tidak boleh null.

2.2 Ontology

Menurut D. Fensel, et al., ontology adalah cara untuk merepresentasikan suatu domain pengetahuan secara eksplisit mengenai suatu konsep dengan cara memberikan makna, property, serta relasi pada konsep tersebut sehingga terhimpun dalam suatu domain pengetahuan dan membentuk suatu basis pengetahuan (Kimball & Ross, 2013). Ontology terbagi atas kelas dan property. Kelas adalah domain utama yang akan digunakan untuk merancang tabel pada data warehouse. Setiap kelas dalam ontology dapat saling berelasi. Relasi ini terjadi karena setiap kelas dalam model konseptual menggunakan ontology merupakan satu kesatuan didalam kelas utama. Sedangkan properties terdiri atas dua jenis, yaitu objek properties dan data properties. Objek properties berisikan relasi yang akan digunakan untuk menghubungkan kelas dan data properties berisikan atribut dari kelas.

2.3 Metode Penelitian

Gambar 1 adalah rancangan umum penelitian yang dilakukan pada penelitian ini untuk menghasilkan *dimensional model* berupa *star schema* menggunakan ontologi sebagai sumber data.



Gambar 1 Rancangan Penelitian

Proses Identifikasi *Data Source* bertujuan untuk mengidentifikasi dan mengambil semua informasi komponen *ontology* yang terdapat pada *data source* berupa *file ontology* yang dimasukkan oleh pengguna. Hal ini diperlukan agar *tool* dapat menerjemahkan informasi tersebut kepada *user*, sehingga mempermudah *user* untuk mengetahui dan menganalisis komponen yang terdapat pada *ontology*. *Tool* mengidentifikasi *data source ontology* dengan membaca model *ontology* untuk mengambil *class*, *property*, dan *individual* dari *ontology*. Jika tidak ditemukan *class* pada *ontology* yang digunakan maka tidak ada hasil yang ditampilkan *tool*. Ketika *ontology* memiliki *class*, maka akan dilakukan proses identifikasi untuk menemukan *data property*, *object property*, dan *individual* dari *class* tersebut. Proses identifikasi ini akan dilakukan untuk setiap *class* yang terdapat pada *file ontology*. Identifikasi pertama yang dilakukan adalah memeriksa *data property* dan *object property* dari sebuah *class*. Apabila tidak ditemukan *data property* dan *object property* maka akan dilanjutkan dengan mengidentifikasi *individual*. Namun apabila *class* memiliki *data property* dan *object property* maka proses pengidentifikasian *property* ini akan dilakukan berulang hingga semua *property* sudah diperiksa dan diidentifikasi. Kemudian hasil identifikasi dari setiap *class* akan ditampilkan pada *tool* seperti nama *class*, *data property*, *object property*, dan *individual*.

Tahap kedua adalah memperoleh daftar Dimensi dan Atribut. Proses ini bertujuan untuk mengidentifikasi dan mengambil semua informasi komponen *ontology* yang terdapat pada *data source* berupa *file ontology* yang dimasukkan oleh pengguna. Hal ini diperlukan agar *tool* dapat mengidentifikasi *class* yang akan didefinisikan sebagai dimensi. *Tool* mengidentifikasi *data source ontology* dengan membaca model *ontology* untuk mengambil *class* dan *property* dari *ontology*. Jika tidak ditemukan *class* pada *ontology* yang digunakan maka tidak ada hasil yang ditampilkan *tool*. Ketika *ontology* memiliki *class*, maka akan dilakukan proses identifikasi untuk menemukan *data property* dari *class* tersebut. Proses identifikasi ini akan dilakukan untuk setiap *class* yang terdapat pada *ontology*. Identifikasi yang dilakukan adalah memeriksa *property* pada *class* atau *data property* yang didefinisikan sebagai atribut dimensi. Terdapat syarat untuk mendefinisikan *class* sebagai dimensi yaitu *class* harus memiliki minimal satu *data property*. Apabila tidak ditemukan *data property* maka *class* tersebut tidak didefinisikan sebagai dimensi dan akan dilanjutkan pengidentifikasian *class* berikutnya. Namun apabila *class* memiliki *data*

property maka proses pengidentifikasian *property* ini akan dilakukan berulang hingga semua *property* sudah diperiksa dan diidentifikasi. Kemudian hasil identifikasi dari setiap *class* akan ditampilkan pada *tool* seperti nama *class* sebagai nama dimensi, nama *data property* sebagai nama atribut.

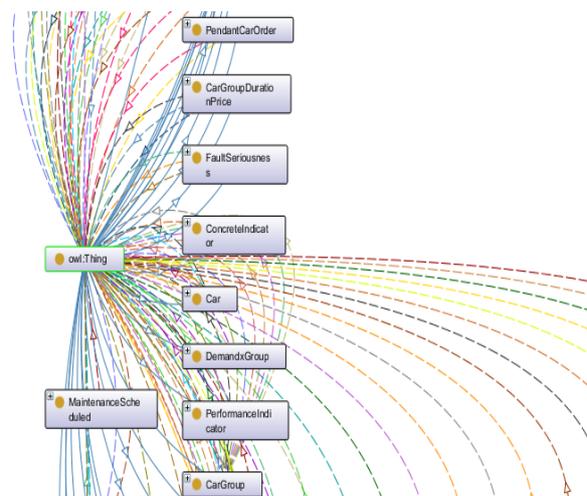
Selain memperoleh daftar Dimensi dan Atribut, daftar Tabel *Fact* juga diidentifikasi. Proses ini bertujuan untuk mengidentifikasi dan mengambil semua informasi komponen *ontology* yang terdapat pada *data source* berupa *file ontology* yang dimasukkan oleh pengguna. Hal ini diperlukan agar *tool* dapat mengidentifikasi *class* yang akan didefinisikan sebagai *fact*. *Tool* mengidentifikasi *data source ontology* dengan membaca model *ontology* untuk mengambil *class* dan *property* dari *ontology*. Jika tidak ditemukan *class* pada *ontology*

yang digunakan maka tidak ada hasil yang ditampilkan *tool*. Ketika *ontology* memiliki *class*, maka akan dilakukan proses identifikasi untuk menemukan *data property* dari *class* tersebut. Proses identifikasi ini akan dilakukan untuk setiap *class* yang terdapat pada *ontology*. Untuk setiap *class* sebagai kandidat *fact tabel* pada *ontology* akan dilakukan pengidentifikasian *property*. Untuk *property* sebagai data *property* akan diperiksa apakah data *property* tersebut bertipe data numerik maka dapat digunakan sebagai *measure event* pada *fact tabel*. Sedangkan *property* yang merupakan *object property*, maka akan diperiksa apakah memiliki *class* yang berelasi sebagai dimensi dari *fact tabel*. Untuk setiap *class* yang berelasi maka akan dilakukan pemeriksaan jumlah atribut yang dimiliki, apabila *class* yang berelasi melalui *object property* memiliki minimal satu *data property*, maka *class* tersebut didefinisikan sebagai dimensi dari *fact tabel*. Kemudian akan ditambahkan id sebagai *natural key* dan *key* sebagai *surrogate key* dan menjadi *primary key* untuk dimensi *fact* tersebut. Kemudian *key* akan ditambahkan sebagai *foreign key* pada *fact tabel*. *Class* sebagai kandidat *fact tabel* akan diperiksa apakah memenuhi syarat pendefinisian *fact tabel* yaitu harus memiliki minimal satu atribut *measure* dan berelasi dengan minimal 2 dimensi. Apabila syarat terpenuhi, maka akan diberikan *fact id*. Kemudian akan ditambahkan tabel dimensi *date* sebagai dimensi yang berelasi kepada setiap tabel *fact* yang telah diidentifikasi. Tabel dimensi *date* ini dapat digunakan sebagai *conformed dimension* serta membantu menentukan tingkat atau level *grain* dari sebuah *fact*. Proses ini membantu *user* untuk dapat mengidentifikasi tabel *fact* yang dapat digunakan sebagai komponen *dimensional model*. Kemudian hasil proses ini seperti nama *class* sebagai nama *fact* dan nama dimensi, nama data *property* sebagai atribut *fact* dan atribut dimensi ditampilkan kepada user.

Tahap terakhir adalah *Generate Fact and Dimension SQL Query*. Setelah tabel *fact* dan dimensi yang berelasi dengannya diidentifikasi, maka

tabel *fact* dan dimensi tersebut diubah ke dalam bentuk kueri agar dapat dilihat struktur datanya setelah dipetakan dari *data source ontology* menjadi tabel *fact* dan dimensi yang akan digunakan sebagai *dimensional model*, serta dapat dieksekusi pada MySQL.

Data yang digunakan oleh peneliti adalah domain EUCar-Rental. Potongan komponen domain *ontology* EUCar-Rental yang dapat dilihat pada Gambar 2. *Class* pada *ontology* adalah suatu set, koleksi, konsep, jenis objek, atau jenis benda. Beberapa *class* pada domain ini memiliki *subclass*. *Ontology* EUCar-Rental memiliki 59 *class*, terdiri dari 28 *class* (level 1), 10 *subclass* (level 2), 12 *subclass* (level 3), 8 *subclass* (level 4), 1 *subclass* (level 5). *Object property* pada *ontology* digunakan untuk menghubungkan *class* dengan *class* lainnya, dengan kata lain *object property* menggambarkan relasi antar *class* dengan *class* lainnya, *class* dengan *subclass*, maupun *subclass* dengan *subclass*. Terdapat 110 *object property* yang menghubungkan antar *class* maupun *subclass* pada *ontology* EUCar-Rental. Data *property* merupakan atribut dari suatu *class*. Terdapat 86 data *property* pada *ontology* EUCar-Rental.

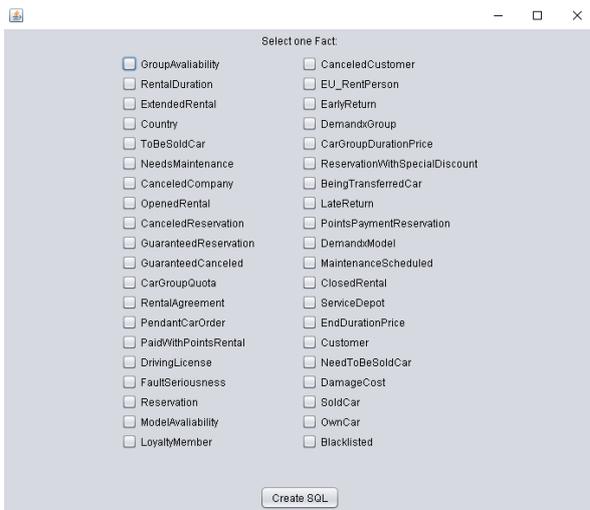


Gambar 2 Ontology EU Rental (sumber: <https://www.cs.upc.edu/~%20oromero/EUCarRental.owl>)

3. HASIL DAN PEMBAHASAN

Transformation tool yang dihasilkan menerima data *source ontology* sebagai input kemudian menampilkan secara detail daftar komponen data *source* yang terdapat pada *file ontology* yang di-input. Kemudian sistem akan menampilkan daftar struktur atau komponen *ontology*. Sistem kemudian akan menampilkan daftar dimensi dan atribut yang dihasilkan pada layar Show Possible Fact Table List. Setelah dim diperoleh, sistem kemudian menampilkan semua *fact* yang dihasilkan dari data *source ontology* yang dimasukkan. *Fact* yang dipilih kemudian akan dibutuhkan untuk proses selanjutnya. Pada Gambar 3 merupakan tampilan dari hasil implementasi *select fact and dimension*.

Kimball seperti pada Gambar 5. Pengujian dilakukan terhadap semua *fact* dan dimensi yang terbentuk.



Gambar 3 Hasil Implementasi Generate Fact and Dimension into SQL Query.

Transformation tool akan melakukan proses perubahan atau transformasi tabel *fact* beserta beberapa tabel dimensi yang berelasi dengannya menjadi kueri SQL, dimana kueri yang dihasilkan ini akan dijadikan sebagai bahan untuk melakukan evaluasi. Saat user memilih *button* Create SQL, maka sistem akan memberikan notifikasi apabila *fact* berhasil di-generate ke dalam kueri. Kueri tersebut kemudian akan di basis data dan akan menghasilkan dimensional model seperti pada Gambar 4. Skema tersebut merupakan hasil akhir dari toos yang dihasilkan pada penelitian ini.

Evaluasi dilakukan dengan menggunakan pengujian terhadap kesesuaian dimensional model yang terbentuk dengan aturan yang ditetapkan oleh

Form Evaluasi Dimensional model
 Domain Ontology:
 Tabel *fact* yang dihasilkan:

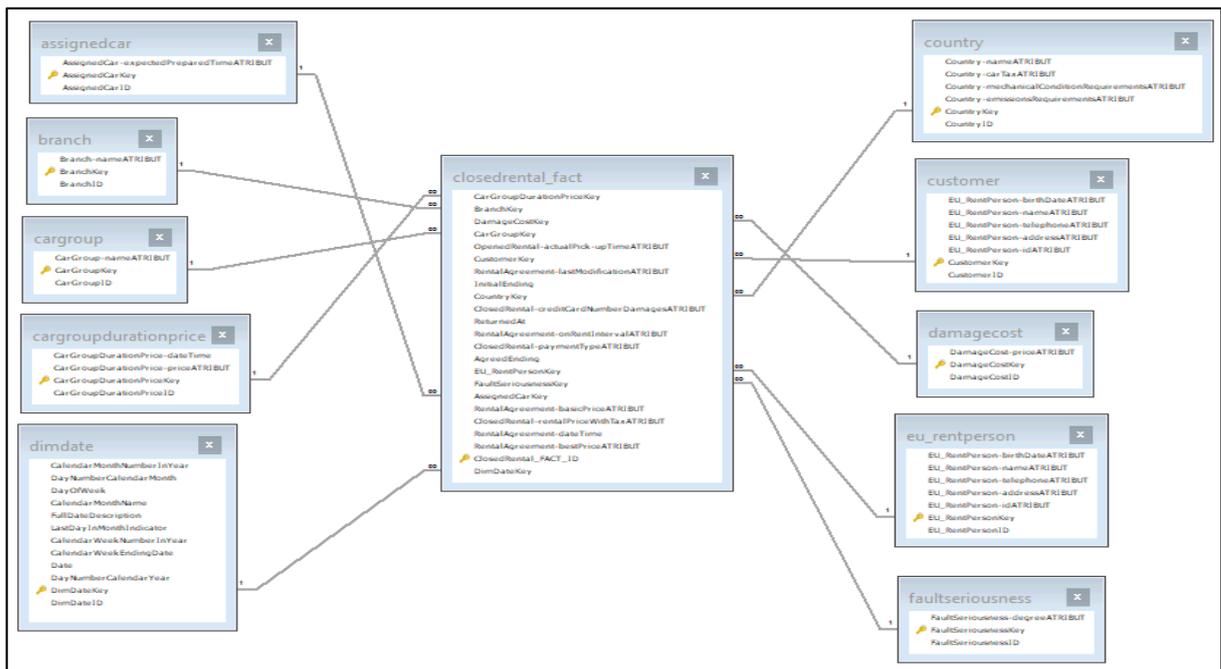
No.	Indikator Dimensional Model	Terpenuhi	Tidak Terpenuhi
1.	Tabel dimensi memiliki <i>surrogate key</i> sebagai <i>primary key</i>		
2.	Key dalam tabel dimensi sederhana dan <i>unique</i>		
3.	Dimensi memiliki <i>date time dimension</i>		
4.	Tabel dimensi memiliki <i>natural key</i> yang dapat bertahan lama		
5.	Tabel <i>fact</i> memiliki atribut bertipe data numerik sebagai <i>measure event</i>		
6.	Tabel <i>fact</i> memiliki <i>transaction, periodic, dan accumulating snapshot fact</i>		
7.	Tabel <i>fact</i> memiliki <i>foreign key</i>		

Gambar 5 kueri dari hasil implementasi generate SQL query

Berdasarkan proses ini, terdapat 40 tabel *fact* atau skema *dimensional model* yang dihasilkan. Setelah melakukan evaluasi, dari 40 *dimensional model* komponen *fact* dan dimensi yang dihasilkan sudah sesuai dengan pendekatan Kimball karena sudah memiliki *surrogate key* sebagai *primary key* pada dimensi, *fact* berelasi dengan tabel dimensi *date/ time*, *fact* memiliki *foreign key* dari *primary key* dimensi yang berelasi dengannya, dan setiap *fact* yang dihasilkan memiliki minimal satu atribut bertipe data numerik serta *fact* berelasi minimal dengan 2 tabel dimensi. Sehingga semua indikator pada form evaluasi terpenuhi.

Berdasarkan skema dimensional model pada Gambar 4 *ClosedRental Fact* setiap tabel dimensi memiliki *surrogate key* yang digunakan sebagai *primary key* dan menjadi *foreign key* pada tabel *fact*, dan terdapat tabel dimensi *date* yang digunakan sebagai *conformed dimension*

Indikator Dimensional Model yang digunakan untuk mengevaluasi hasil adalah menggunakan



Gambar 4 Dimensional Model ClosedRental

checklist untuk melihat apakah persyaratan yang ditetapkan telah terpenuhi atau tidak. Persyaratan tersebut adalah : tabel dimensi memiliki surrogate *key* sebagai *primary key* , *key* dalam tabel dimensi sederhana dan unique, dimensi memiliki date/ time dimension, tabel dimensi memiliki natural *key* yang dapat bertahan lama, tabel *fact* memiliki atribut bertipe data numerik sebagai measure event, tabel *fact* memiliki transaction, periodic, dan accumulating snapshot *fact* , tabel *fact* memiliki foreign *key* . Pengujian yang dilakukan terhadap dimensi dan *fact* tabel yang terbentuk menunjukkan bahwa tools yang sudah dapat menghasilkan dimensional model yang sesuai dengan pendekatan Kimball.

4. KESIMPULAN

Kesimpulan yang diperoleh dari penelitian ini adalah bahwa penelitian ini sudah mencapai tujuan penelitian, dimana proses pembuatan *dimensional model* dengan sumber data *ontology* telah dilakukan dan menghasilkan tabel *fact* dan dimensi yang membentuk *dimensional model* yang sesuai dengan pendekatan *dimensional model*.

Adapun yang menjadi kekurangan dalam penelitian ini adalah dimensional model yang dihasilkan belum dapat ditampilkan secara visual pada tools yang dihasilkan karena dimensional model akan terlihat setelah kueri yang dihasilkan dijalankan di basis data.

DAFTAR PUSTAKA

- ALEKSIC, S., CELIKOVIC, M. & LINK, . S., 2010. Faceoff: Surrogate vs. Natural Keys. ADBIS 2010, p. 543–546.
- BOJIĆIĆ, I. et al., 2016. A Comparative Analysis of Data Warehouse Data Models of Data Warehouse Data Models. 2016 6th International Conference on Computers Communications and Control (ICCCC).
- FEKI, J. & HACHAICHI, Y., 2013. An Automatic Method For The Design Of Multidimensional Schemas From Object Oriented Databases. International Journal of Information Technology & Decision Making, Volume 12, pp. 1223-1259.
- G, B., POŠČIC, P. & JAKŠIC, D., 2017. Data Warehouse Architecture Classification. MIPRO.
- KIMBALL & ROSS, 2013. Kimball Dimensional Modeling Techniques Overview”, “The Data Warehouse Toolkit : The Definitive Guide to Dimensional Modeling”. s.l.:John Wiley & Sons Inc..
- KIRMANI, M. M., 2017. Dimensional modeling using Star Schema for Data Creation. Oriental Journal of Computer Science and Technology, Volume 10, pp. 745-754.
- LI, J., XIAO, M., SUN, Y. & CHEN, . Y., 2021. Knowledge Modeling of Airborne Missile Management System Based on Multi-dimensional Fuzzy Ontology. Journal of Physics: Conference Series.
- MOULAI, H. & DRIAS, H., 2018. From Data Warehouse to Information Warehouse: Application to Social Media. New York, International Conference on Learning and Optimization Algorithms: Theory and Applications.
- REN, S. & WANG, T., 2018. Dimensional Modeling of Medical Data Warehouse Based on Ontology Shuxia. 2018 IEEE 3rd International Conference on Big Data Analysis, ICBD 2018.
- RUS, A. M. M. & OTHMAN, Z. A., 2019. ONTODB : Aplikasi untuk Transformasi Ontologi OWL ke Basis Data Relasi SQL. Journal of Data Analysis, 1(2).
- SAUTOT, L., BIMONTE, S. & JOURNAUX, L., 2015. A semi-automatic design methodology for (Big) Data Warehouse transforming facts into dimensions. JOURNAL OF IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.
- SCHOLTZ, I., 2016. Inmon versus Kimball: The agile development of a data warehouse, Potchefstroom: North-West University.
- SEBAA, A., CHIKH, F., NOUICER, A. & TARI, A., 2018. Medical Big Data Warehouse: Architecture and System Design, a Case Study: Improving Healthcare Resources Distribution. Transactional Processing Systems.
- THENMOZHI, M. A. V. K., 2013. A Tool for Data Warehouse Multidimensional Schema Design using Ontology. International Journal of Computer Science I, p. 161–169.
- WAMARS, H. L. H. S. & RANDRIATOAMANANA, R., 2016. Datawarehouse: A Data Warehouse artist who have ability to understand data warehouse schema pictures. 10 Conference (TENCON) -.
- YESSAD, L. & LABIOD, A., 2016. Comparative Study of Data Warehouses Modeling Approaches: Inmon, Kimball and Data Vault. International Conference on System Reliability and Science.