

Penerapan *Hybrid Slowly Change Dimension* Untuk *Nearly Realtime Datawarehouse*

Ni Wayan Wisswani

Politeknik Negeri Bali, Bali
e-mail: wisswani@yahoo.com

Abstrak

Datawarehouse yang bersifat *nearly realtime* membutuhkan pemodelan dimensi dan fakta secara *realtime*. Pemodelan dimensi sangat penting untuk dilakukan karena dimensi akan menjadi sumber bagi fakta. Teknik pemodelan dimensi yang akan diterapkan pada makalah ini adalah *hybrid slowly change dimension*. Teknik ini akan menciptakan beberapa *field* baru untuk menampung perubahan yang mungkin saja terjadi pada *database* sumber saat manipulasi terjadi sehingga fakta tidak kehilangan informasi. Dalam usaha untuk mengimplementasikan dimensi yang bersifat *nearly realtime* maka *hybrid slowly change dimension* akan dikembangkan dengan metode *change data capture*. Hal ini akan menangkap setiap perubahan data yang dapat mempengaruhi dimensi yang terjadi pada *database* sumber dan kemudian mentransformasikannya sehingga dapat sesuai dengan *hybrid* dimensi yang telah didesain.

Kata kunci: *nearly realtime datawarehouse, hybrid slowly change dimension, change data capture*

Abstract

Datawarehouse that is nearly realtime requires the model of table dimension and the fact in realtime. Modeling the dimension is very important because the dimension will be a source for facts. This research used such kind of dimensional modeling technique is a hybrid slowly change dimension. This technique will create a new field to accommodate the changes of source database when the manipulation is executed so that facts don't lose the information. In order to implement this dimension therefore hybrid slowly change will be deployed using change data capture method. This method will record any changes of data in source database that may affect the dimensions and then will be transformed them so that agree with the hybrid that has been designed.

Keywords: *nearly realtime datawarehouse, hybrid slowly change dimension, change data capture*

1. Pendahuluan

Kebutuhan organisasi untuk melakukan analisa data dan pembuatan laporan secara cepat dan terintegrasi dari *online transaction processing (OLTP)* mengakibatkan *data warehouse* yang dikembangkan dengan konsep *nearly realtime datawarehouse (NRTDWH)* menjadi penting untuk dikembangkan [1]. Namun ETL sebagai inti proses [2,3] dalam *data warehouse* yang mengelola data secara *time variant* tidak mampu melakukan prosesnya agar menghasilkan *datawarehouse* yang bersifat *nearly real time* [4]. Dalam usaha untuk menghasilkan *NRTDWH*, ETL dapat menerapkan metode *change data capture (CDC)* dalam implementasinya [2]. Teknik ini akan dapat mengetahui setiap perubahan pada sumber data dan menangkapnya untuk di-load oleh ETL ke dalam *database* tujuan [5,6].

Perubahan yang berhasil ditangkap oleh metode CDC akan mempengaruhi *dimensional modelling* yang didesain pada *NRTDWH* baik pada tabel dimensi dan fakta [7]. Perubahan pada tabel fakta akan terjadi lebih cepat, sementara tabel dimensi akan berubah perlahan dalam kurun waktu yang lebih lama [8]. Tabel fakta berubah melalui peningkatan jumlah baris, namun tabel dimensi tidak hanya mengalami perubahan dalam jumlah baris, tetapi juga melalui

perubahan atribut [6]. Perubahan tabel dimensi sangat penting untuk dikelola karena dimensi merupakan tabel yang akan menjadi referensi bagi tabel fakta [2].

Untuk mengelola perubahan pada tabel dimensi yang dihasilkan oleh *change data capture*, maka pencatatan tabel ini akan dikelola dengan metode *slowly changing dimension* (SCD) [9]. SCD memiliki beberapa tipe dan salah satunya adalah *hybrid slowly change dimension*. Tipe ini akan melakukan pencatatan atas nilai data yang lama serta data baru sehingga NRTDWH tidak akan kehilangan *history* data yang pernah tercatat dalam tabel dimensi [2]. Metode ini penting untuk dikembangkan karena hilangnya *history* data dalam tabel dimensi sebagai tabel sumber akan menurunkan kualitas informasi yang akan dihasilkan NRTDWH, karena akan terjadi sekumpulan fakta yang tidak bisa dijelaskan oleh fakta setelah dilakukan *joint* dengan tabel dimensi. Berdasarkan pemaparan di atas maka pada makalah ini akan mengulas mengenai penerapan implementasi *hybrid slowly change dimension* pada tabel dimensi agar mampu menghasilkan NRTDWH.

2. Kajian Pustaka

2.1 Nearly Realtime Data Warehouse

Menurut [6] *real time datawarehouse* berbeda dengan tradisional *datawarehouse*. *Datawarehouse* tradisional bersifat pasif, menyediakan data yang bersifat *history*, sedangkan *realtime datawarehouse* bersifat dinamis, dimana *datawarehouse* ini akan menyediakan data yang selalu *up to date*, sehingga data yang dihasilkan merupakan data terkini yang didapatkan secara terus menerus dengan waktu tunggu yang hampir mendekati nol.

Sementara itu menurut [4] bahwa *realtime datawarehouse* dikerjakan dengan sistem yang tidak pernah mati sehingga proses *loading* data dari data *source* tidak pernah berhenti, karena bila proses berhenti maka akan terjadi perbedaan antara data yang telah mengalami perubahan dengan informasi yang dihasilkan. ETL tradisional dapat dimodifikasi dengan mengatur *query* untuk mengurangi waktu periode *load* data agar dihasilkan *real time datawarehouse* atau *near real time datawarehouse*.

2.2 Dimensional modelling

Pemodelan dimensional merupakan konsep desain yang banyak digunakan untuk mengembangkan suatu *datawarehouse*. Model dimensional tersebut terdiri dari struktur data yang diperlukan untuk merepresentasikan dimensi serta fakta dari proses bisnis yang ada. Dalam menggambarkan relasi *database* pada *datawarehouse* digunakan 2 model pendekatan yang disebut model skema yaitu skema bintang (*star schema*) dan skema *snowflake* [2].

2.3 Komponen dimensional modelling

Tabel fakta. Dalam pemodelan dimensional menurut [7], tabel fakta terdiri atas *measurement*, *metric*, atau fakta dari proses bisnis yang ada. Ciri-ciri dari tabel fakta adalah sebagai berikut:

1. *Primary key* pada tabel fakta terdiri atas gabungan lebih dari satu *primary key* yang dimiliki tabel-tabel dimensi yang terkait (*concatenated key*).
2. Memiliki tingkatan data yang telah teridentifikasi.
3. Mudah untuk melakukan rekap data.
4. Memiliki jumlah *record* yang banyak.
5. Memiliki kolom atau atribut yang sedikit.
6. Tidak memiliki *row* yang berisi nilai *null*.
7. Memiliki *degenerated dimension*.

Tabel dimensi. Dalam pemodelan dimensional, tabel dimensi menggambarkan karakteristik keadaan dari *measurement* atau *metric* yang ada [5]. Ciri-ciri dari tabel dimensi adalah sebagai berikut:

1. Memiliki *key* unik pada tabel dimensi (*primary key*).
2. Memiliki jumlah kolom atau atribut yang banyak.

3. Atributnya *textual* dan tidak saling berhubungan.
4. Tabelnya tidak dilakukan normalisasi.
5. Mempunyai kemampuan untuk *drill-down* dan *roll-up*.
6. Memiliki jumlah *record* yang sedikit dibandingkan tabel fakta.

2.4 Slowly Changing Dimension (SCD)

Teknik SCD digunakan untuk mencatat perubahan lambat yang terjadi pada tabel dimensi agar *history* data yang tersimpan dalam tabel dimensi tidak hilang [2]. Terdapat beberapa tipe SCD diantaranya :

Tipe 1 SCD. Tipe ini akan membentuk *record* baru untuk menggantikan *record* lama, dimana hanya ada satu *record* pada *database* sebagai *current data*.

Tipe 2 SCD. Tipe ini akan membentuk *record* baru yang ditambahkan pada tabel dimensi, sehingga terdapat dua *record* pada *database* berupa *current record* dan *record*/data sebelum data mengalami perubahan.

Tipe 3 SCD. Tipe ini akan memodifikasi data yang aslinya dengan memasukkan informasi baru di dalamnya. Hal ini menyebabkan pada *database* terdapat satu *record* yang akan mengandung data yang lama dan ditambah data sebagai informasi baru pada baris yang sama.

2.5 Hybrid SCD

Teknik ini mengkombinasikan semua tipe SCD dalam satu *record* dimensi. Pada teknik ini akan ditambahkan kolom untuk menampung nilai *field* lama serta nilai *field* yang baru setelah terjadi perubahan. Teknik ini juga akan menambahkan kolom untuk menampung waktu *effective date* terjadinya perubahan.

2.6 Change Data Capture

CDC (*changed data capture*) menurut [10] dirancang untuk memaksimalkan efisiensi dari proses ETL. Tanpa CDC semua data yang ada pada ODS akan dipindahkan ke *data warehouse* kapanpun dibutuhkan, sementara dengan CDC hanya perubahan-perubahan data yang terjadi pada ODS saja yang akan dipindahkan. Oleh karena itu CDC dapat meminimumkan *restore* yang digunakan untuk memindahkan perubahan pada data dan meminimalkan waktu *latency* pengiriman informasi bisnis kepada konsumen sehingga tentu saja hal ini dapat menghemat biaya.

3. Metodologi Penelitian

3.1 Ruang Lingkup

Makalah ini akan membahas *hybrid slowly change dimension* yang diterapkan dengan *change data capture* untuk dapat menghasilkan dimensi yang bersifat *nearly realtime data warehouse*.

3.2 Metode

Berikut ini adalah metode yang akan dilakukan dalam penelitian ini, diantaranya :

1. Analisa *metadata technical* dan *business* untuk mengetahui hubungan antara OLTP dan *field* pada DWH.
2. Desain tabel dimensi yang bertipe *hybrid slowly change dimension*.
3. Mendesain proses *change data capture* untuk mendapatkan perubahan yang terjadi pada OLTP dan mencatat pada tabel dimensi.
4. Pengujian dan analisa hasil.

4. Analisa dan Pembahasan Data

4.1 Analisa OLTP

OLTP yang akan digunakan sebagai studi kasus dalam penelitian ini adalah sistem disertasi dan sistem tesis Universitas Udayana. Dalam kedua sistem ini terdapat beberapa tabel yang dikelola OLTP, namun tabel yang menjadi sumber bagi *datawarehouse* adalah tabel prodi dan tabel disertasi yang berasal dari sistem disertasi serta tabel prodi dan tabel tesis dari sistem tesis.

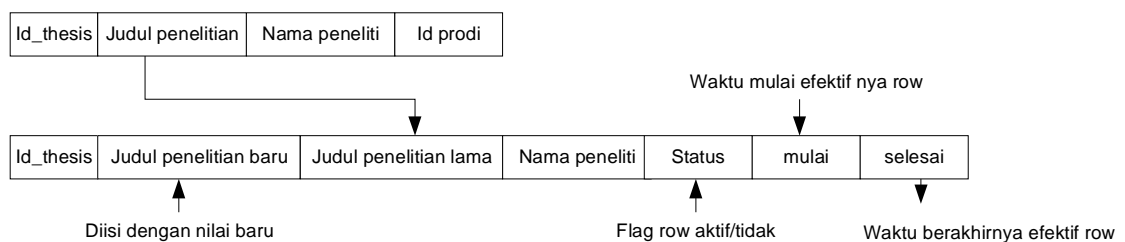
Keempat tabel tersebut akan menjadi sumber bagi NRTDWH yang akan diimplementasikan. Tabel prodi akan menjadi sumber dari tabel dimensi prodi, sedangkan beberapa *field* dari tabel tesis dan disertasi akan menjadi dimensi tesis dan disertasi. Pada Tabel 1 berikut ini merupakan *metadata* yang akan menghasilkan *hybrid slowly Change dimension*.

Tabel 1. Metadata

OLTP sumber	Tabel sumber	Nama field	Tabel tujuan pada DWH
Sistem Tesis	Tabel tesis	Id_thesis	Dimensi tesis
Sistem Tesis	Tabel tesis	judulpenelitian	Dimensi tesis
Sistem Tesis	Tabel tesis	namapeneliti	Dimensi tesis
Sistem Tesis	Tabel tesis	id_prodi	Dimensi tesis
Sistem Tesis	Tabel prodi	Id_prodi	Dimensi prodi
Sistem Tesis	Tabel prodi	Id_jenis	Dimensi prodi
Sistem Tesis	Tabel prodi	Nama_prodi	Dimensi prodi
Sistem disertasi	Tabel disertasi	Id_disertasi	Dimensi disertasi
Sistem disertasi	Tabel disertasi	Judul_disertasi	Dimensi disertasi
Sistem disertasi	Tabel disertasi	Nama_peneliti	Dimensi disertasi
Sistem disertasi	Tabel disertasi	id_prodi	Dimensi disertasi
Sistem disertasi	Tabel prodi	Id_prodi	Dimensi prodi
Sistem disertasi	Tabel prodi	Id_jenis	Dimensi prodi
Sistem disertasi	Tabel prodi	Nama_prodi	Dimensi prodi

4.2 Desain tabel dimensi dengan *Hibrid Slowly Change Dimension*

Berdasarkan 4 tipe SCD yang ada, dalam penelitian ini akan diterapkan *hybrid SCD*. Teknik ini akan mencatat perubahan yang terjadi dalam sebuah kolom barupada tabel dimensi. Nilai kolom lama yang mengalami perubahan akan digunakan untuk menampung *field* dan sebelum berubah dan kolom baru akan diisi nilai dari perubahan yang dilakukan. Untuk menentukan *row* data yang aktif maka *row* data akan diberi penanda sertadibuat kolom *field* waktu saat *record* valid dan *field* waktu terakhir saat *record* tidak valid. Dengan cara ini *update* data pada OLTP baik berupa *insert*, *update* dan *delete* tidak akan menghilangkan *history* data yang telah tersimpan dalam *nearly realtime datawarehouse*. Gambar 1 berikut ini adalah desain SCD yang akan diterapkan dalam *nearly real time data warehouse*.



Gambar 1. Desain *slowly change dimension*

Desain SCD tersebut akan diimplementasikan dalam tabel-tabel dimensi dengan hasil seperti berikut ini

a. Dimensi Prodi sistem *thesis*

Tabel 2. Tabel dimensi prodi dari sistem *thesis*

Field	Keterangan
Id_st_prodi	<i>Surrogate key</i> , merupakan <i>primary key</i> dari tabel ini
Id_prodi	Menerangkan id dari prodi yang diambil dari tabel th_prodi
Nama_prodi	Menerangkan nama prodi yang diambil dari tabel th_prodi
Mulai	Mulaiefektif berlakunya <i>field</i>
Selesai	Selesaiefektif berlakunya <i>field</i>
Nm_prodilama	Namaprodi sebelumnyayang sudahberubahakibat proses <i>update</i>
Status	Menandakan status prodi yang sedangaktif, bila status 1 maka artinya <i>field</i> tersebut dalam kondisi aktif, sedangkan 0 maka <i>field</i> tersebut dalam kondisi tidak aktif

b. Dimensi *thesis* pada sistem *thesis*

Tabel 3. Tabel dimensi *thesis* dari sistem *thesis*

Field	Keterangan
Id_st_tesis	<i>Surrogate key</i> , <i>primary key</i> dari tabel ini
Id_tesis	Menerangkan id dari <i>thesis</i> yang diambil dari tabel th_tesis
Judul_penelitian_lama	Menerangkan judul penelitian sebelum proses edit yang diambil dari tabel th_tesis
Judul_penelitian_baru	Menerangkan judul penelitian setelah proses edit yang diambil dari tabel th_tesis
Nama_peneliti	Menerangkan nama peneliti yang diambil dari tabel th_tesis
Mulai	Mulaiefektif berlakunya <i>field</i>
Selesai	Selesaiefektif berlakunya <i>field</i>
Status	Menandakan status judul <i>thesis</i> , status 1 untuk status yang sedang aktif, sedangkan 0 untuk status yang tidak aktif
Id_prodi	Diambil dari <i>surrogate key</i> tabel dimensi prodi

c. Dimensi disertasi sistem disertasi

Tabel 4. Tabel dimensi disertasi dari sistem disertasi

Field	Keterangan
Id_sd_disertasi	<i>Surrogate key</i> , yang diciptakan untuk menjadi <i>primary key</i>
Id_disertasi	Kode disertasi yang dicatat diambil dari tabel th_disertasi pada ODS disertasi
Judul_penelitian	Judul peneliti diambil dari tabel th_disertasi
Nama_peneliti	Nama peneliti yang diambil dari tabel th_disertasi
Status	Status dari judul disertasi digunakan untuk mencatat perubahan status disertasi, status 0 melambangkan bahwa judul tersebut tidak lagi aktif, sementara status 1 melambangkan judul tersebut sedang aktif.
Mulai	Mulaiefektif berlakunya <i>field</i>
Selesai	Selesaiefektif berlakunya <i>field</i>
Jdl_lama	Judul penelitian yang lama sebelum dilakukan <i>update</i>
Id_prodi	<i>Field</i> ini merupakan <i>surrogate key</i> id prodi pada dimensi prodi

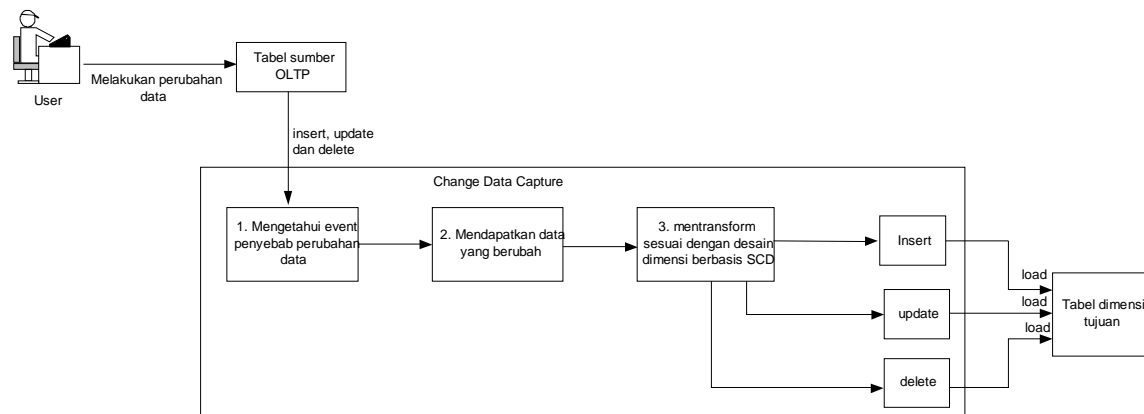
d. Dimensi prodi sistem disertasi

Tabel 5. Tabel dimensi prodi dari sistem disertasi

Field	Keterangan
Id_sd_prodi	<i>Surrogate key</i> , <i>primary key</i> tabel ini
Id_prodi	Id prodi yang diambil dari tabel th_prodi pada sistem disertasi
Nama_prodi	Nama prodi yang diambil dari tabel th_prodi

Mulai	Mulaiefektif berlakunya <i>field</i>
Selesai	Selesaiefektif berlakunya <i>field</i>
Nm_prodi_lama	Nama prodi sebelumnya yang sudah berubah akibat proses <i>update</i> .
Status	Menandakan status prodi yang sedang aktif/terbaru, 1 status aktif 0 untuk status tidak aktif

Agar dihasilkan dimensi yang dapat menghasilkan NRTDWH maka dimensi *hybrid slowly change dimension* akan diterapkan dengan metode CDC. Perubahan pada tabel sumber akibat *event insert, update* maupun *delete* akan diketahui oleh CDC. Hasil *capture* kemudian ditransformasikan ke dalam bentuk yang sesuai dengan desain tabel dimensi yang telah dimodelkan dengan *hybrid SCD*, dan kemudian akan di-*load* ke dalam tabel dimensi bersesuaian. Metode ini akan bekerja dengan alur seperti Gambar 2 berikut ini.



Gambar 2. Desain umum proses *nearly realtime hybrid Slowly change Dimension*

4.3 Pengujian *Hybrid SCD*

Dalam pengujian ini akan dilakukan manipulasi pada proses di OLTP yang mempengaruhi DWH. Untuk poses *insert*, pengujian dilakukan dengan memasukkan data Prodi Teknik Elektro melalui *form* pada sistem disertai seperti Gambar 3 berikut ini .

Gambar 3. *Form* untuk melakukan proses *insert*

Data yang di-*input* akan disimpan kedalam tabel *th_prodi*, hal ini terlihat dengan telah masuknya data teknik elektro ke dalam tabel pada sistem *existing* yang terlihat pada Gambar 4 berikut ini.



Gambar 4. Form hasil insert

Proses simpan yang dilakukan ke dalam tabel `th_prodi` akan memicu CDC untuk melakukan `capturefield` `id_prodi`, `Nama_prodi`, dan waktu saat proses `insert` terjadi. Hasil `capture` tersebut akan di-load ke dalam tabel dimensi prodi sebagai row data baru. Row data baru ini akan berisi `fieldid_sd_prodi` yang merupakan `surragate key`, `id_prodi`, `Nama_prodi`, dan waktu saat proses `insert` terjadi dan pemberian nilai 1 pada `field status` yang menyatakan bahwa row tersebut sedang aktif. Hasil kerja CDC akan mempengaruhi tabel dimensi prodi sehingga tampak seperti Gambar 5 berikut ini :

id_sd_prodi	id_prodi	nama_prodi	mulai	selesai	nm_prodi_lama	status
108032012001	35	teknik elektro	2012-03-08 16:44:34	(NULL)	(NULL)	1

Gambar 5. Nilai tabel hasil insert

Event lain yang akan menyebabkan tabel dimensi prodi berubah adalah `update` pada tabel `th_prodi` yang dilakukan melalui form seperti Gambar 6 berikut ini :



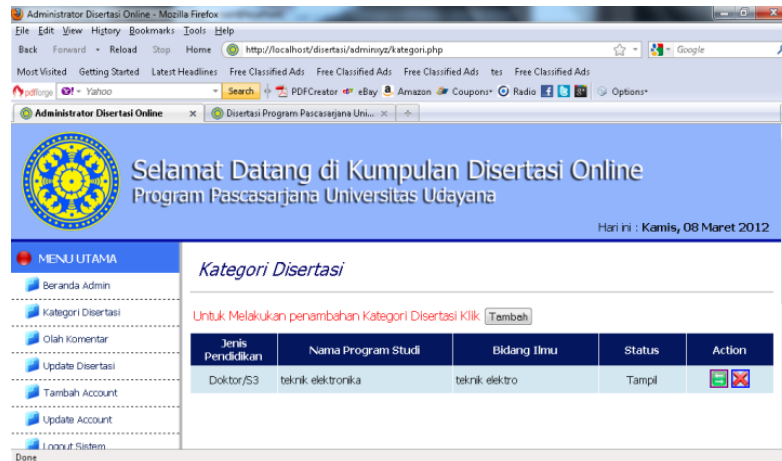
Gambar 6. Form untuk melakukan proses update

Perubahan yang dilakukan pada tabel `th_prodi` akan memicu CDC mencatat nama prodi baru yang di-`insert`, nama prodi yang lama yang di-`update`, serta waktu saat kedua `field` diubah. Hasil `capture` tersebut akan di-`insert`-kan sebagai row baru ke dalam tabel dimensi prodi berupa `fieldid_prodi`, `Nama_prodi`, `mulai`, `status`, `Nm_prodi_lama`, dan `Selesai`. Setelah itu akan dilakukan `updatefieldStatus` dan `fieldSelesai` dimensi prodi pada row data yang memiliki `idprodi` sama dengan `idprodi` baru yang di-load dan masih berstatus 1. Proses ini menyebabkan `fieldStatus` akan diubah menjadi 0 dan `fieldSelesai` berisi waktu saat perubahan dilakukan. Proses CDC pada event ini akan mengubah tabel dimensi prodi sehingga tampak seperti Gambar 7 berikut ini.

id_sd_prodi	id_prodi	nama_prodi	mulai	selesai	nm_prodilama	status
108032012001	35	teknik elektro	2012-03-08 16:44:34	2012-03-08	(NULL)	0
108032012002	35	teknik elektronika	2012-03-08 17:00:20	(NULL)	teknik elektro	1

Gambar 7. Hasil *update* dimensi

Proses *delete* pada tabel *prodi* sistem disertasi juga akan mengubah *status* dimensi *prodi*. Proses ini dilakukan melalui *form* seperti Gambar 8 berikut ini.

Gambar 8. Form *delete* pada sistem OLTP

Saat proses *delete* dilakukan data *field* *Id_prodi* dan waktu proses penghapusan *row* data akan *capture*. Kemudian akan dilakukan *update* terhadap dimensi *prodi* pada *row* data yang *id* *prodi*nya sama dengan hasil *capture* dan memiliki *Status* 1. *Update* dilakukan dengan membuat *field* *status* menjadi 0 dan *field* *selesai* diisi oleh waktu saat proses *delete* dilakukan. Proses CDC pada *event* ini akan menyebabkan tabel dimensi *prodi* berubah seperti hasil yang tampak pada Gambar 9 berikut ini

id_sd_prodi	id_prodi	nama_prodi	mulai	selesai	nm_prodilama	status
108032012001	35	teknik elektro	2012-03-08 16:44:34	2012-03-08	(NULL)	0
108032012002	35	teknik elektronika	2012-03-08 17:00:20	2012-03-08	teknik elektro	0

Gambar 9. Hasil dimensi saat proses *delete*

5. Simpulan

Pencatatan perubahan tabel dimensi dalam *data warehouse* harusnya tidak menghilangkan *history* data yang pernah tersimpan didalamnya untuk tetap mempertahankan kualitas data. Agar seluruh *history* perubahan data yang mempengaruhi tabel dimensi tersimpan sehingga informasi yang hilang dapat diminimalkan maka diimplementasikan *hybrid slowly change dimension*. Agar *Hibrid SCD* yang didesain dapat menampung data dalam *nearly realtime data warehouse* maka diterapkan metode *change data capture*.

Daftar Pustaka

- [1] Simitsis A, Vassiliadis P, Sellis T, "Optimizing ETL Processes in Data Warehouses In Data Engineering", Proceedings 21st International Conference on Digital Object, pp: 564 – 575, 2005. <http://citeseerx.ist.psu.edu> [diunduh : 10 Agustus 2011]

- [2] Kimball Ralph, Caserta Joe, The Data Warehouse ETL Toolkit Practical Techniques for Extraction, Cleaning, Conforming and Delivering Data, Canada: Wiley Publishing. Inc, 2004.
- [3] Savitri F.N, Laksmiwati H, "Study of localized data cleansing process for ETL performance improvement in independent datamart", Electrical Engineering and Informatics (ICEEI), International Conference, 2011. [diunduh: 13 Agustus 2011]
- [4] Langseth Justin, "Real-Time Data Warehousing: Challenges and Solutions", 2004. <http://dssresources.com/papers/features/langseth/langseth02082004.html>. [diunduh: 12 Agustus 2011]
- [5] Mitchell J Eccles, David J Evans and Anthony J Beaumont, "True Real-Time Change Data Capture With Web Service Database Encapsulation", IEEE 6th World Congress on Services, 2010. [diunduh: 10 Agustus 2011]
- [6] Ponniah, Paulraj, "Data warehousing fundamentals for IT professionals / 2nd ed", John Wiley & Sons. Inc, 2010.
- [7] Inmon, W.H, "Building The Data Warehouse", Fourth Edition, Canada: Wiley Publishing. Inc, 2005.
- [8] Avignon France, "Chapter 5 Advances in Database Technology EDBT '9", 5th International Conference on Extending Database Technology, March 25-29, 1996. [diunduh: 11 Agustus 2011]
- [9] Santos V, Belo O, Sch. of Manage. & Technol., Porto Polytech., Felgueiras, Portugal, Information Systems and Technologies (CISTI), 6th Iberian Conference, Juni 2011. [diunduh : 12 Agustus 2011]
- [10] Ankorian, Itamar, "Information Management Magazine", January 2005. <http://www.information-management.com/issues/20050101/1016326-1.html> [diunduh: 12 Agustus 2011]