ISSN 2087-2658
Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)
Volume 5, Nomor 2, Juli 2016

Penerapan *Extraction-Transformation-Loading* (ETL) Dalam Data Warehouse (Studi Kasus : Departemen Pertanian)

Rahmadi Wijaya
Prodi D3 Teknik Informatika, Fakultas Ilmu Terapan
Universitas Telkom
Email : rakit2272@yahoo.com

Bambang Pudjoatmodjo
Prodi D3 Teknik Informatika, Fakultas Ilmu Terapan
Universitas Telkom
Email : b.pudjoatmodjo@gmail.com

ABSTRAK
Proses *Extraction-Transformation-Loading* (ETL) pada pembangunan data warehouse berperan melakukan ekstraksi data dari berbagai sumber, pengubahan data ke bentuk yang sesuai dengan kebutuhan dan pengisian ke *storage data warehouse*.

Termasuk di dalam fungsi ETL terdapat proses data *cleansing* yang menangani redundansi, inkonsistensi dan integritas data. Proses ETL akan mengisikan data dari sumber ke *integration layer* yang merupakan tempat data dari *data warehouse* tergabung dan terintegrasi. Dari *integration layer*, data dapat dikelompokkan dengan lingkup yang lebih kecil dan sesuai kebutuhan dalam *repository* lain yang disebut *data mart*. Program *reporting* dari *data warehouse* akan berhubungan dengan *data mart* sebagai sumber datanya.

Penelitian ini merupakan bagian dari Penelitian berbasis profesi pembangunan *data warehouse*. Secara khusus Penelitian menangani proses ETL. Pada Penelitian ini dibangun metadata bagi proses ETL yang menyimpan definisi data sumber, data arget (*data warehouse*), dePenelitian keterhubungan antara data sumber dan data target, serta dePenelitian transformasi data dari data sumber ke data target. Dibangun juga program ETL dengan teknologi DTS pada SQL Server 2000 yang menggunakan metadata tersebut. Diharapkan dengan adanya metadata bagi proses ETL dapat dibangun program ETL yang memiliki tingkat *reusability* tinggi.

Kesimpulan utama yang dapat diambil dari Penelitian ini adalah penggunaan proses ETL yang dinamis (menggunakan metadata ETL) sangat diperlukan apabila berhubungan dengan sistem operasional yang masih labil dan berkemungkinan besar mengalami perubahan skema basis data. Proses ETL dinamis juga diperlukan untuk menangani kebutuhan pengguna akan laporan yang bertambah.

Kata Kunci : *Data Warehouse*, ETL, *Integration Layer*, *Reusability*, Metadata

1. Pendahuluan

Kunci sukses bagi sebuah perusahaan untuk bertahan pada masa sekarang adalah kemampuan untuk menganalisa, merencanakan dan bereaksi terhadap perubahan lingkungan bisnis secara cepat dan akurat. Kemampuan ini hanya dapat dipenuhi dengan tersediannya informasi yang memadai bagi para manajer, direktur dan para pengambilan keputusan lainnya. Informasi yang dibutuhkan tersebut seringkali adalah berupa data operasional dan sulit untuk didapatkan. Walaupun tersedia akses ke data tersebut, seringkali format dan struktur data yang ada tidak seperti yang diinginkan. Teknologi *Data warehouse* adalah sebuah konsep dan perangkat yang memungkinkan penyediaan akses ke seluruh level informasi perusahaan. *Data warehouse* memungkinkan suatu organisasi untuk mengumpulkan data dari berbagai format dan standar yang berbeda, melakukan analisa atas data tersebut, dan mengeluarkan report-report yang

dibutuhkan bagi para analis dan pengambilan keputusan.

Data warehouse memiliki fungsi ETL (*extraction, transformation, loadingI*) yang berperan dalam pengisian data untuk kebutuhan analisis dengan melakukan ekstraksi dari berbagai sumber data, pengubahan data ke bentuk yang sesuai dan pengisian ke *storage data warehouse*. Termasuk di dalam fungsi ETL terdapat proses *data cleansing* yang menangani redundansi, inkonsistensi dan integritas data. Proses ETL akan mengisikan data dari sumber ke *integration layer* yang merupakan tempat data dari *data warehouse* tergabung dan terintegrasi. Dari *integration layer*, data dapat dikelompokkan dengan lingkup yang lebih kecil dan sesuai kebutuhan dalam r*epository* lain yang disebut *data mart*. Program *reporting* dari *data warehouse* akan berhubungan dengan *data mart* sebagai sumber datanya.

Kondisi Departemen Pertanian pada saat ini dalam melakukan analisis dan pengolahan data sebagai berikut :

1. Setiap sistem informasi memiliki basis data masing-masing yang berbeda-beda bahkan masih ada yang dalam bentuk *flat file* (.xls).
2. Data tersebar di setiap sistem informasi dan ada data yang mengalami redundansi.
3. Dalam analisis dan pengambilan keputusan, pihak eksekutif Departemen Pertanian harus melakukan rekap ulang dari data setiap sistem informasi.

Dalam meningkatkan kinerja departemen, para eksekutif Departemen Pertanian perlu dibekali dengan data yang akurat dan terkini unuk menghasilkan keputusan-keputusan strategis. Tulisan dalam Penelitian ini akan mengangkat tentang studi kasus perencanaan *data warehouse* pada Departemen Pertanian. Pembahasan akan difokuskan pada siklus pengembangan *data warehouse*, desain dan arsitektur *data warehouse*, khususnya yang berkaitan dengan informasi basis data statistik Pertanian dan Basis data Ekspor Impor pada Departemen Pertanian.

Dengan menggunakan *data warehouse*, diharapkan pihak eksekutif Departemen Pertanian dapat mengakses data yang mereka butuhkan tanpa harus bergantung pada laporan yang diberikan oleh masing-masing bagian. Selain mempercepat akses data, *data warehouse* dikembangkan ini juga diharapkan dapat menyelesaikan masalah redundansi dan inkonsistensi.

1. Data Warehouse

1.1. Arsitektur Data warehouse

Data warehouse adalah suatu sistem dengan arsitektur yang bersifat terbuka, jadi untuk membangun suatu arsitektur *Data warehouse* sangat tergantung pada kebutuhan sistem (*system requirements*). Gambar 2.1 menunjukkan salah satu contoh dari arsitektur *Data warehouse*. Arsitektur tersebut menunjukkan perbedaan penggunaan *Data warehouse* dibandingkan dengan basis data operasional. Data dalam *Data warehouse* digunakan untuk keperluan spesifik, umumnya berupa analisis oleh aplikasi-aplikasi dalam *Executive Information System* (EIS) atau *Decision Support System* (DSS). Sedangkan basis data operasional umumnya digunakan oleh aplikasi-aplikasi transaksional yang mampu melakukan *read/write* terhadap basis data tersebut.

Arsitektur data warehouse dibangun berdasar kebutuhan tertentu, hal ini dapat kita lihat dari hal berikut :

1. Data input bagi *Data warehouse* tidak lagi hanya berasal dari sistem internal (sumber operasional pada umumnya), melainkan dirancang untuk dapat mengakomodasi sumber eksternal (data dari luar sistem operasional), misalnya :
   - Data dari bursa efek,

- Data dari internet (dengan teknologi *web farming*),
- Data dari sistem *mobile* (misalnya *phone cell*).

2. Informasi yang tersimpan dalam *Data warehouse* dapat dispesialisasikan lagi menjadi beberapa *Data warehouse* yang lebih khusus (*Data Mart*) sehingga dalam arsitektur terdapat proses tambahan untuk mempopulasikan data dari *Data warehouse* ke dalam beberapa *Data Mart.*
3. Aplikasi yang berada pada layer pengguna berkembang menjadi beberapa model misalnya : berbasis web, berbasiskan desktop, ataupun berbasiskan sistem *mobile.*

Data warehouse mengumpulkan beberapa data mentah dari sistem operasional, dan mungkin terdapat juga data eksternal lainya, kemudian melakukan operasi terhadap data tersebut (membersihkan, mengintegrasikan, dan menyimpan data), dan akhirnya mendistribusikan hasilnya ke pengguna. Beberapa kriteria dalam mendesain arsitektur *data warehouse* adalah sebagai berikut [COR01] :

1. Melakukan ekstraksi data dari bermacam sumber.
   Data warehouse biasanya melibatkan lebih dari satu data sumber.
2. Mengintegrasikan data ke tempat yang umum.
   Setelah data diekstrak dari beberapa sumber, data tersebut tidak bisa begitu saja dimasukkan ke *data warehouse*. Data tersebut biasanya perlu dibersihkan dan dilakukan pengecekan terhadap keterhubungan antar data untuk kemudian diintegrasikan ke basis data yang sama.
3. Menyimpan data dalam format yang dapat digunakan pengguna.
   Data dari berbagai sumber, yang sebenarnya mewakili data sejenis, mungkin mempunyai format yang berbeda. Untuk dapat digunakan, data tersebut harus mengalami konversi menjadi format yang dimengerti oleh pengguna.
4. Menyediakan mekanisme bagi pengguna agar dapat mengakses *data warehouse*.
   Pengguna *data warehouse* bisa saja membutuhkan sudut pandang baru yang lebih kompleks untuk mendukung perubahan analisis perusahaan. Untuk itu, *data warehouse* harus memiliki mekanisme fleksibel yang dapat memenuhi perubahan kebutuhan pengguna akan akses terhadap basis data.

1.2. Akses Pengguna

Akses pengguna merupakan komponen yang mendefinisikan bagaimana pengguna dapat melakukan akses pada *Data warehouse.* Pada umumnya untuk melakukan akses terhadap *Data warehouse,* pengguna menggunakan aplikasi/perangkat lunak untuk mendefinisikan *query* dan analisis. Perangkat lunak tersebut disebut sebagai *Business Inteligence Software*. Sebagai contoh dari *Business Inteligence Software* adalah *:*

1. Sistem Pendukung Keputusan (*Decision Support System*) memungkinkan untuk melakukan *ad hoc query* dan membangun report yang dapat digunakan untuk mendukung pengambilan keputusan.
2. Sistem Informasi Eksekutif (*Executive Information Systems*) merupakan kombinasi antara *Decision Suport System*/DSS dengan kemampuan untuk melakukan analisis dan akses terhadap sumber eksternal (misal layanan informasi saham dari Dow Jones).
3. OLAP (*Online Analytical Processing*) memungkinkan melakukan analisis terhadap data operasional sehingga bisa menentukan trend

bisnis,pendeteksian kesalahan terhadap proses bisnis, dan sebagainya.

4. *Data Mining Tools* memungkinkan melakukan otomasi analisis terhadap data untuk menemukan pola-pola tertentu yang dapat digunakan untuk melakukan penyesuaian di dalam proses bisnis.

2. ETL (Extraction, Transformation, Loading)

Di dalam proses ETL, data dari berbagai sumber secara periodik akan diekstrak dan diintegrasikan ke dalam *data warehoue* [ONE97]. *Extraction* merupakan proses untuk mengidentifikasi seluruh sumber data yang relevan dan mengambil data dari sumber-sumber tersebut. *Transformation* adalah proses yang mempunyai peran dalam melakukan pembersihan data dan integrasi skema yang berbeda-beda ke dalam skema yang terdefinisi dalam *data warehouse*. *Loading* adalah proses pemindahan data secara fisik dari system operasional ke dalam *data warehouse*.

2.1. Konsep ETL

Pendefinisian lingkup ETL dengan cara menganalisis tiap target tabel (dimensi dan fakta) perlu dilakukan pada awal pembangunan arsitektur proses ETL. Untuk tiap tabel target, perlu diambil informasi tentang kelakuannya, dimanakah data sumbernya dan proses bisnis apa saja yang bergantung kepadanya.

2.1.1. Metadata

Pada awal perkembangan *data warehouse*, proses integrasi direalisasikan dengan membuat program ETL spesifik untuk struktur basis data sumber dan basis data *data warehouse*.

Kemudian, ditemukan bahwa program-program ETL spesifik tadi pada intinya melakukan proses yang seragam. Banyak blok-blok program yang bisa dipergunakan kembali untuk proses ETL lainnya. Pada titik itulah dikembangkan teknologi ETL *tools* yang dapat melakukan otomasi proses integrasi data ke *data warehouse*.

Secara singkat, metadata adalah data tentang data. Khususnya, metadata ETL adalah data tentang proses ETL. Pendefinisian metadata ETL diperlukan dalam rangka membangun program ETL yang mempunyai tingkat *reusability* tinggi.

2.1.2. Extraction

Proses ektraksi mengidentifikasi seluruh sumber data yang relevan dan seefisien mungkin mengambil data tersebut. Program ekstraksi berjalan melalui file atau basis data, menggunakan berbagai kriteria dalam memilih data, dan menemukan data yang sesuai, kemudian mentransportasikan data ke file atau basis data lainnya.

Change data capture (CDC) adalah elemen penting dalam analisis ekstraksi. Transaksi yang dijadikan data fakta hampir selalu mempunyai timestamps. Tetapi, data dimensi pada system sumber tidak selalu mempunyai timestamps karena kecenderungannya yang tidak bergantung pada suatu event. Oleh sebab itu, CDC paling sulit diimplementasikan dengan data dimensi.

Ada beberapa cara mengimplementasikan CDC. Jika pada basis data sumber atau file terdapat *timestamps*, implementasi akan jauh lebih mudah. Bila tidak, CDC bisa diimplementasikan dengan membaca log file dari basis data sumber atau mengimplementasikan trigger pada basis data sumber. Pengimplementasian seperti ini sangat riskan karena terlalu bergantung kepada teknologi yang dimiliki basis data sumber. Implementasi seperti ini tidak akan berlaku jika data sumber berupa file.

Beberapa cara dalam mengenali perubahan pada basis data sumber :

1. *Timestamp*
   Ekstraksi pada sistem yang menyimpan timestamp terhadap waktu *insert* dan *update* record, membuat CDC tidak perlu melakukan pencarian ke seluruh isi

tabel untuk mengenali record apa saja yang telah berubah.

2. *Trigger*
   Trigger diimplementasikan pada tabel sumber. Setiap record yang disimpan, diubah ataupun dihapus, trigger akan menuliskan pesan ke log file. Log file inilah yang akan dijadikan informasi bagi *data warehouse* untuk meng-update datanya. Pada rakteknya, implementasi trigger jarang dilakukan, karena membutuhkan modifikasi terhadap sistem sumber yang berkemungkinan menurunkan performasi dari sistem sumber.
3. *File Compare*
   Pembandingan dilakukan antara data yang terdapat pada sistem sumber sekarang dengan data terakhir yang dimiliki oleh *data warehouse*. Teknik ini kurang akurat dibandingkan teknik-teknik sebelumnya, karena teknik ini menggunakan metode pembandingan periodik snapshots.

2.1.3. Transformation

Transformation adalah proses manipulasi terhadap data dari sistem sumber ke format lain pada *data warehouse* atau *data mart* dalam rangka menjadikannya sebuah informasi yang bermakna. Fungsi-fungsi transformasi yang mungkin dilakukan adalah: [COR01]

1. Konversi Format
   Data harus dikonversi menjadi format yang umum.
2. Manipulasi *string*
   Meliputi konkatenasi, *trim, up case, lower case* dan sebagainya.
3. Fungsi aritmatik
   Proses aritmatik bisa dibuat dalam modul terpisah ataupun menggunakan fungsi aritmatik yang dimiliki oleh SQL.
4. *Conditional Assignment*
   Biasanya diimplementasikan menggunakan tabel *lookup* yang menyimpan nilai lama dari data dan mendefinisikan nilai barunya.
5. Agregasi
   Agregasi akan menghasilkan data dengan tingkat detail tertentu.
6. *Conditional Branch*
   Mirip dengan *conditional assignment* tetapi fungsi kondisional pada *conditional branch* lebih rumit dengan tingkat kondisional yang bertingkat. Perlu lebih dari sekedar tabel *lookup* untuk mengimplementasikannya.
7. *Referential Integrity*
   Transformasi harus menjaga integritas dari *data warehouse*, bahkan walaupun bersumber dari data tidak valid yang tidak terjaga integritasnya.
8. *Surrogate Key Resolution*
   Surrogatye key memberikan fleksibilitas tinggi dalam menangani data pada *data warehouse* dibandingkan penggunaan *key* bawaan data sumber.
9. *User Written*
   Pengguna bisa mendefinisikan fungsi-fungsi tersendiri yang merupakan gabungan dari fungsi aritmatik dan fungsi kondisional sesuai dengan proses bisnis perusahaan.

2.1.4. Loading

Proses loading akan memindahkan data yang telah ditransformasi ke *data warehouse*. Strategi *loading* ke *integration layer* dibagi menjadi dua bagian, yaitu strategi *loading* bagi table dimensi dan strategi *loading* bagi tabel fakta [COR01].

A. Strategi *Loading* Tabel Dimensi

Ada tiga strategi *loading* tabel dimensi. Setiap strategi menangani perubahan data dimensi secara berbeda bergantung kepada cara penanganan terhadap data dimensi lama. Pada ketiga strategi tersebut, jika tidak ditemukan sebuah *record* dengan *natural key* yang sama, maka *record* baru akan ditambahkan. *Natural key* adalah atribut pada dimensi yang secara unik membedakan *record* dimensi (bukan

surrogate key). Jika ditemukan *record* dengan *natural key* yang sama, maka :

Strategi 1 :
Sejarah data tidak disimpan pada strategi 1. Jika nilai *input* telah ada di tabel dimensi berdasarkan nilai *natural key*, maka *record* tersebut akan diperbaharui.

Strategi 2 :
Critical column adalah kolom-kolom penting pada tabel dimensi yang tetap dijaga keberadaannya. Jika pada *input* ditemukan satu atau lebih kolom yang termasuk *critical column* dan nilai pada input berbeda dengan nilai pada kolom bersesuaian pada tabel dimensi (dan tetap berdasarkan nilai *natural key*), maka *record* yang bersesuaian tadi akan *expired* dan *record* baru dengan *surrogate key* yang baru akan dimasukkan. Jika tidak ditemukan kesamaan nilai pada *critical column*, maka seperti pada strategi 1, *record* yang bersesuaian akan diperbaharui.

Strategi 3 :
Hampir mirip dengan strategi 2 yang melakukan penanganan terhadap perubahan nilai pada *critical column*. Bedanya, pada strategi 3 untuk tiap *critical column*, beberapa kolom yang berbeda ditempatkan pada tiap *record* untuk menyimpan nilai data sekarang dengan n nilai data sebelumnya. Ketika terjadi perubahan nilai pada *critical column*, semua nilai akan digeser, dan nilai terakhir (nilai paling lama) akan dihapus, kemudian nilai terbaru dimasukkan.

1. Ilustrasi Kasus

ESS diimplementasikan pada Departemen Pertanian yang memiliki banyak bagian yang bergerak di berbagai bidang. Penggunaan EIS yang memanfaatkan *data warehouse* bertujuan untuk mengintegrasikan data dari berbagai sumber basis data yang tersebar di bagian produksi untuk meningkatkan efesiensi dan efektifitas pengambilan keputusan oleh pihak eksekutif.

Pada Departemen Pertanian, tidak terdapat basis data terintegrasi antar bagian. Basis data dikelola secara lokal dan terpisah untuk bagian, dengan masing–masing bagian menerapkan format data yang tidak seragam. Hal ini akan mempersulit pihak eksekutif, yaitu pihak yang menduduki jabatan di pusat, dalam hal pengambilan keputusan yang sifatnya global. Data yang tersebar ini akan menyulitkan pihak eksekutif dalam melakukan analisis untuk mengekstrak data menjadi informasi yang diperlukan.

Selain itu, basis data umumnya digunakan untuk menyimpan data yang terperinci. Hal ini kurang sesuai dengan kebutuhan data yang datang dari pihak eksekutif. Pihak eksekutif umumnya menginginkan data yang berupa rangkuman, dan dalam bentuk tampilan grafis berupa diagram untuk memudahkan dalam pengambilan keputusan.

Adapun permasalahan yang mungkin dihadapi dalam mengimplementasikan EIS pada Departemen Pertanian, berkaitan dengan banyaknya bagian yang dimiliki adalah :

1. Antara satu basis data dengan basis data lainnya ada kemungkinan terdapat inkompatibilitas data, hal ini disebabkan oleh tidak diterapkannya format standar untuk data.
2. Mengekstrak informasi. Dari berbagai data yang tersimpan pada *data warehouse*, sebuah EIS dituntut untuk dapat mengekstrak data mentah dari berbagai sumber yang tersedia, untuk kemudian menghasilkan informasi yang berguna untuk pihak eksekutif, biasanya untuk memprediksi *trend* untuk lima tahun ke depan.
   Untuk mengekstraksi informasi, perlu ditentukan format *request* informasi kemudian dicocokkan dengan data yang terdapat pada *data warehouse*. Sebagai ilustrasi, misalkan informasi yang diminta adalah berupa informasi umum dari seluruh bagian yang dipegangnya.

Misalkan format data yang telah ditentukan untuk *request* informasi tersebut adalah berupa : jumlah pengeluaran pada bulan tersebut, segmentasi *customer*, biaya operasional per bulan, serta alokasi biaya resiko.

3. Menjaga akurasi data. Dalam hal ini, timbul masalah bagaimana memastikan data yang diakses oleh pengguna EIS adalah data yang diinginkan.
4. Menentukan selang waktu yang tepat untuk memproduksi dan menyimpan data ke dalam *data warehouse*. Misalkan menentukan frekuensi pengumpulan data dari seluruh bagian, apakah satu bulan, dua bulan, atau selang waktu yang lain.

   Menentukan batas waktu untuk menyimpan data. Mengingat jumlah data yang tersimpan bisa mencapai jumlah yang sangat banyak, oleh karena itu perlu ditentukan batas waktu penyimpanan data. Agar menghemat kapasitas memori dan meningkatkan performansi EIS dari segi kecepatan.

2. Analisis

Pada bagian analisis akan dibahas mengenai batasan masalah dalam pembangunan proses ETL, skema *integration layer* yang digunakan, proses ETL yang dipilih, analisis kebutuhan metadata, analisis pembangunan proses ETL dinamis dan teknologi yang berkaitan dengan pembangunan proses ETL dinamis.

Pada analisis ini juga akan menganalisis konsep-konsep umum dalam proses ETL. Proses ETL yang dimaksud adalah proses ETL antara data sumber dan *integration layer*. Proses ETL akan dibangun di atas basis data sumber dan *data warehouse* yang menggunakan skema relasional. *Proses* ETL dinamis dan metadata nya yang akan dibangun terbatas terhadap pemenuhan kebutuhan kasus pembangunan *data warehouse* Departemen Pertanian.

1.1. Analisis Skema Integration Layer

Skema *integration layer* yang dibangun akan memenuhi prinsip integritas dan mengabaikan kecepatan respon terhadap *query* dari pengguna. Nantinya pengguna tidak akan melakukan *query* secara langsung ke *integration layer* melainkan ke *data mart*. Untuk memenuhi kebutuhan di atas, skema *integration layer* yang dibangun memenuhi prinsip normalisasi.

Tabel-tabel pada *integration layer* dibagi menjadi dua macam:

- Tabel Dimensi

Tabel dimensi lebih jarang mengalami perubahan dibandingkan dengan tabel fakta. Ekstraksi tabel dimensi melibatkan pengecekan terhadap data dimensi yang telah ada. Apabila data dimensi telah ada sebelumnya, tidak akan dilakukan penambahan ke basis data. Secara berkala akan dilakukan pengecekan kevalidan data dimensi pada basis data *data warehouse* dengan data sumber. Apabila data yang berhubungan sudah tidak ada di data sumber, maka data pada *data warehouse* akan *expired*. Pada data sumber, data dimensi tidak berhubungan dengan *event* tertentu. Jika data dimensi masih valid, maka data dimensi masih terdapat pada data sumber.

- Tabel Fakta

Tabel fakta akan terikat pada *event* tertentu di basis data sumber. Sehingga pada setiap *record* pada tabel fakta akan berhubungan dimensi waktu. Ekstraksi tabel fakta berasumsi data yang telah diekstrak tidak akan berubah lagi nilainya pada basis data sumber. Untuk itu, ekstraksi tabel fakta akan dilakukan jika tabel fakta sumber sudah tidak mungkin dirubah lagi, sehingga data fakta pada *data warehouse* dapat terjamin kevalidannya. Tabel fakta akan diekstraksi pada tabel sumber yang sifatnya transaksional dan

ISSN 2087-2658
Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)
Volume 5, Nomor 2, Juli 2016

relatif cepat mengalami perubahan (penambahan).

6.2. Analisis Proses ETL

A. Extraction

Penentuan pendekatan yang digunakan pada ekstraksi sangat terkait dengan analisis bisnis proses, pendefinisian area subjek, serta desain logik/fisik *data warehouse*. Berdasarkan tiga metode mengenali data mana yang harus diekstrak yaitu *timestamp, trigger,* dan *file compare* akan ditentukan metode mana yang akan diaplikasikan pada pembangunan ETL dinamis.

Metode-metode pengenalan perubahan pada tabel sumber ini, hanya dilakukan terhadap tabel-tabel dimensi. Seperti yang telah disebutkan sebelumnya, ekstraksi tabel-tabel fakta dilakukan dengan asumsi telah dideterminasi sebelumnya *record-record* mana yang harus diekstrak.

Metode *timestamp* sangat bergantung terhadap kondisi sistem sumber. Metode ini bisa diterapkan jika pada sistem sumber yang melakukan pencatatan waktu terhadap pemasukan dan perubahan *record*. Akan lebih mudah lagi jika penghapusan *record* tidak benar-benar menghapus *record* dari basis data, tetapi hanya memberikan *flag* "terhapus" pada *record*.

Dengan metode ini setiap detail perubahan sistem sumber dapat dideterminasi dengan tingkat presisi waktu tinggi. Pengaplikasian metode ini menambah jumlah atribut pada tiap tabel sebanyak minimal dua atribut, yaitu *timestamp insert* dan *flag* valid data. Dengan *timestamp*, proses ekstraksi akan lebih efisien dalam melakukan pencarian *record* mana yang harus diekstrak. Secara umum, pengaplikasian metode ini tidak terlalu berpengaruh terhadap performansi sistem sumber.

Sama seperti metode *timestamp*, metode *trigger* juga sangat bergantung terhadap kondisi sistem sumber. Metode ini bisa diterapkan hanya terhadap sistem sumber yang mendukung teknologi *trigger*. Dengan metode ini, setiap detail perubahan sistem sumber dapat dideterminasi dengan tingkat presisi waktu tinggi. Pengaplikasian metode ini membutuhkan tambahan *storage* untuk menyimpan *log file*. Dengan *trigger*, proses ekstraksi akan lebih efisien dalam melakukan pencarian *record* mana yang harus diekstrak. Pengaplikasian metode ini akan menurunkan performansi sistem sumber, sebab setiap kali transaksi terjadi akan ada proses tambahan pada sistem sumber yaitu proses penulisan ke dalam *log file*.

Metode *file compare* (atau bisa disebut *record compare*) tidak bergantung terhadap kondisi sistem sumber. Dengan metode ini, detail dari perubahan data sumber tidak dapat dideterminasi karena sifatnya yang melakukan perbandingan periodik *snapshots*. Metode ini tidak membutuhkan penambahan *storage* pada sistem sumber. Proses *file compare* merupakan proses rumit yang membutuhkan banyak sumber daya dalam melakukan ekstraksi. Tetapi pengaplikasiannya tidak berpengaruh terhadap performansi sistem sumber.
Matriks perbandingan tiga metode di atas dapat dilihat pada tabel 3.1

Tabel 3.1 Perbandingan Metode Ekstraksi

No.	Aspek Pendukung	Timestamp	Trigger	File Compare
1	Ketergantungan terhadap sistem sumber	Bergantung	Bergantung	Tidak Bergantung
2	Akurasi data terekstrak	Akurat	Akurat	Kurang akurat
3	Penambahan *storage* sistem Sumber	Butuh	Butuh	Tidak Butuh
4	Efisiensi proses	Efisien	Efisien	Kurang Efisien
5	Pengaruh terhadap performansi sistem sumber	Kecil	Besar	Tidak Berpengaruh

Berdasarkan matriks perbandingan metode ekstraksi, maka dipilih *file compare* sebagai metode ekstraksi proses ETL dinamis yang akan dibangun. Alasan utamanya adalah metode *file compare* tidak bergantung terhadap sistem sumber sedangkan pada kasus pembangunan *data warehouse* Departemen Pertanian, tidak dimungkinkan mekanisme perubahan sistem sumber.

1.3. Transformation

Berdasarkan kebutuhan pada pembangunan *data warehouse* Departemen Pertanian maka fungsi-fungsi transformasi yang dipilih untuk diimplementasikan adalah konversi format, manipulasi string, fungsi aritmatik, *conditional assignment*, *referential integrity* dan *surrogate key resolution*.

Format masukan data yang didukung oleh proses ETL dinamis adalah *string, number* dan *date*. *String* dan *date* akan disimpan pada *integration layer* sebagai *string*. Sedangkan *number* tetap akan disimpan sebagai *number*.

Manipulasi string yang dilakukan adalah operasi string sederhana, yaitu *up case* dan *lower case* yang diimplementasikan pada *query* pengambilan data sumber.

Fungsi aritmatik yang digunakan terbatas pada operasi aritmatik sederhana yaitu *sum* untuk penjumlahan, *count* untuk penghitungan jumlah *record* dan *avg* untuk menghitung nilai rata-rata data sumber.

Conditional assignment akan diimplementasikan menggunakan tabel transformasi yang menyimpan nilai data sumber beserta nilai transformasinya. Fungsi *conditional assignment* ini tidak memikirkan aspek waktu, sehingga kondisi transformasi nilai akan valid pada saat proses ETL dilakukan saja.

Referential integrity dijaga dengan asumsi data sumber telah terjaga integritasnya, sehingga proses ETL hanya melakukan ekstraksi terhadap *referential integrity* yang telah didefinisikan pada data sumber.

Surrogate key diterapkan dengan tujuan sebagai pembeda bagi nilai data dimensi yang mungkin memiliki nilai *natural key* yang sama dalam kurun waktu valid yang berbeda.

1.4. Loading

Berikut ini akan dibahas strategi *loading* bagi tabel dimensi dan tabel fakta. Seperti telah disebutkan sebelumnya, ada tiga strategi loading tabel dimensi yaitu :

(1) Melakukan *update* terhadap *record* bersesuaian.
(2) Mengimplementasikan *critical column* dengan penambahan *record*.
(3) Mengimplementasikan *critical column* dengan penambahan sejarah isi pada *record* yang sama.

Strategi pertama memiliki kelemahan dalam nilai keakuratan data. Sebab, bisa ditemukan suatu data dimensi dengan nilai *natural key* yang sama tetapi mengacu pada informasi yang berbeda pada jangka waktu yang berbeda. Di lain pihak, strategi pertama membutuhkan *storage* yang paling sedikit diantara strategi yang lain.

Strategi dua memiliki keakuratan penyimpanan nilai paling tinggi di antara strategi lainnya. Data dimensi dengan nilai *natural key* yang sama dan jangka waktu keberlakuan yang berbeda, tetap dapat disimpan dengan resolusi *surrogate key*. Data dari tabel fakta tetap dapat akurat mengacu ke data dimensi yang seharusnya. Di lain pihak, strategi kedua ini membutuhkan *storage* yang paling banyak diantara strategi yang lain.

Strategi tiga selain sulit diterapkan (tidak ada query yang mampu melakukan pergeseran nilai antar atributnya) juga kurang memiliki nilai keakuratan. Penyimpanan sejarah nilai atribut tidak akan banyak berharga tanpa penyimpanan nilai waktu keberlakuannya.

Untuk menjamin keakuratan data maka strategi dua akan dipilih sebagai metode *loading* pada proses ETL dinamis yang akan dibangun. Pada proses ETL yang dibangun, tidak akan disimpan nilai-nilai *critical column* untuk tiap tabel dimensi, sehingga tiap-tiap kolom pada

ISSN 2087-2658
Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)
Volume 5, Nomor 2, Juli 2016

tabel dimensi akan dianggap sebagai *critical column*.

Seperti yang sudah dijelaskan sebelumnya, proses *loading* ke tabel fakta akan dilakukan tanpa memperhatikan data sejarah di tabel fakta. *Loading* ke tabel fakta akan berasumsi data yang akan dimasukkan belum pernah dimasukkan sebelumnya ke tabel fakta.

1.5. Analisis Metadata Proses ETL Dinamis

Informasi metadata yang dipilih akan sangat berpengaruh terhadap proses ETL dinamis yang akan dibangun. Tabel 3.2 memperlihatkan jenis-jenis metadata apa saja yang dibutuhkan untuk membangun sebuah proses ETL dinamis.

Tabel 3.2 Jenis Metadata Proses ETL

No.	*Item* Metadata	Didapatkan dari Proses
1	Definisi data sumber	Analisis sumber. Skema basis data sumber ataupun definisi dari *File* sumber.
2	Definisi data target	Proses pemodelan. Pemodelan skema basis data pada *data warehouse* ataupun *data mart* yang mendukung kebutuhan analisis.
3	DePenelitian bisnis antara data pada sumber dan target	Analisis sumber, proses pemodelan, atau proses ETL. Jika data tidak tersedia pada skema sumber ataupun *tool* pemodelan, data harus didefinisikan secara manual pada ETL *tool*.
4	Metoda dan pengelolaan hak pengaksesan data	Definisi *middleware*, *username* dan *password* untuk mengakses data sumber dan target.
5	Transformasi	Proses ETL
6	Pembedaan jenis tabel (table dimensi atau fakta)	Analisis sumber.
7	*Data lineage* (dePenelitian tentang semua transformasi yang terjadi pada data sumber untuk mendapatkan *domain* sumber)	Proses ETL
8	Statistik ETL (seperti, Waktu terakhir proses ETL, jumlah *record* ter-*load*, jumlah *error*, dan lain-lain)	Proses ETL

1.6. Analisis Pembangunan Proses ETL Dinamis

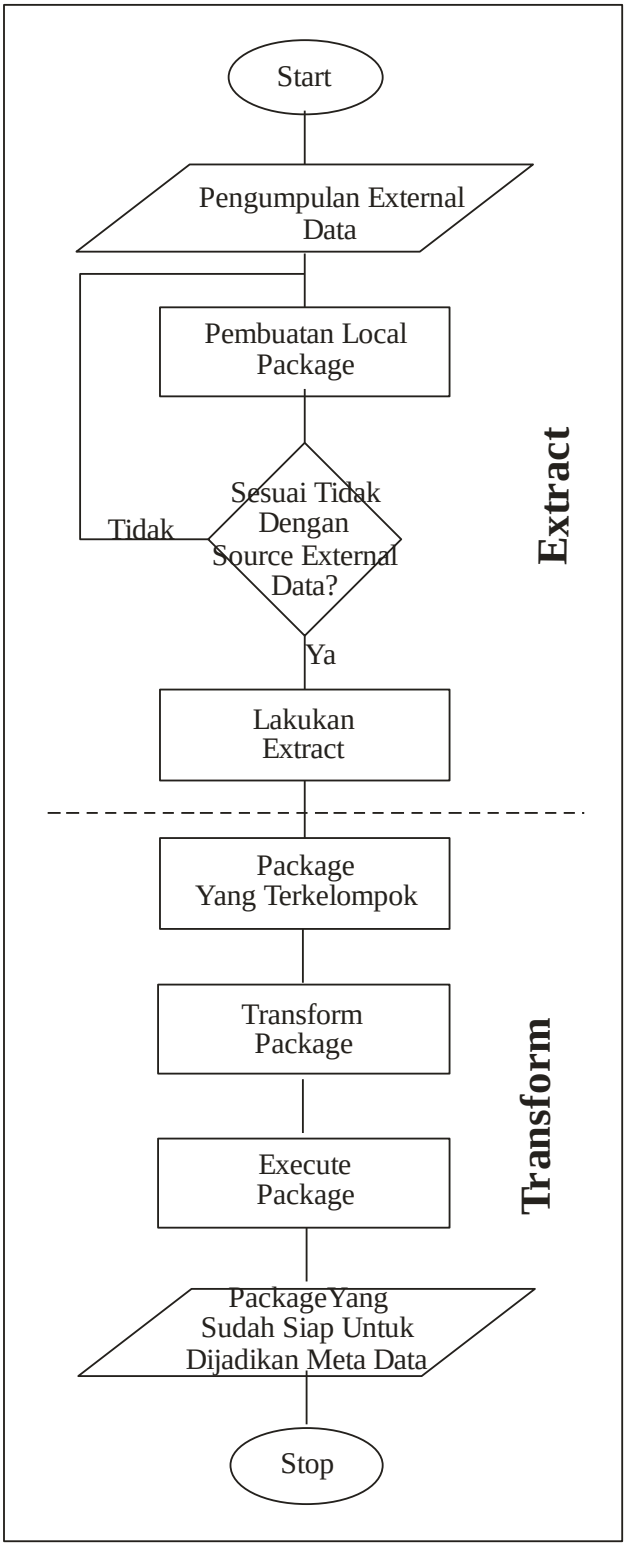
Pembangunan proses ETL dinamis akan terbatas terhadap pemenuhan kebutuhan pembangunan *data warehouse* Departemen Pertanian. Proses ini harus mampu independen terhadap skema data sumber, skema *integration layer*, bisnis proses antara data pada data sumber dan *integration layer* dan proses transformasi yang didefinisikan pada bisnis proses.

Pendefinisian skema data sumber hanya akan diimplementasikan berupa definisi *query* pengaksesan datanya. Untuk tiap-tiap atribut pada data sumber harus dilakukan *rename* agar sesuai dengan nilai atribut yang bersesuaian pada basis data *integration layer*. Skema *integration layer* akan didefinisikan dalam skema basis data relasional.

Pada skema table *integration layer* akan dibedakan antara tabel dimensi dan tabel fakta. Bisnis proses antara data sumber dan *integration layer* akan didefinikan dalam relasi *one-to-one* antara data dimensi atau fakta dengan data pengaksesan *query* pada data sumber. Proses transformasi akan didefinisikan secara sederhana berupa tabel transformasi yang menyimpan nilai data lama beserta nilai transformasinya.

ISSN 2087-2658
Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)
Volume 5, Nomor 2, Juli 2016

1.7. Algoritma ETL



Gambar 3.1 Algoritma ETL

Proses Extract pada ETL dimulai dengan pengumpulan external data. Dimana external data tersebut dapat diambil dari beberapa source, diantaranya SQL Server 2000, Oracle, Excel, HTML, dsb. Sesudah pengumpulan external data, dilanjutkan dengan pembuatan package (paket, kemudian melakukan pengecekan apakah source external data sudah sesuai. Contoh dalam hal ini adalah source external data dari SQL Server 2000 tidak dapat dijadikan satu dengan source external data Oracle. Apabila source external data belum sesuai, maka proses dikembalikan lagi ke tahap awal, yaitu pembuatan package. Apabila source external data sudah sesuai, maka sistem melakukan extract.

Setelah sistem melakukan extract, maka selesailah sudah proses extract dalam ETL, yang kemudian dilanjutkan dengan proses Transform. Dimana package yang sudah terkelompok, di transformasikan (transform) kemudian di eksekusi (execute) sehingga menghasilkan package yang siap untuk dijadikan metadata.

Proses Loading dalam ETL hanyalah sebatas me-*load* package menjadi sebuah metadata.

1.8. Analisis Teknologi yang digunakan

Sesuai kebutuhan dari pengguna, *data warehouse* akan dibangun pada DBMS SQL Server 2000. Pembangunan proses ETL dinamis dapat menggunakan beragam teknologi. Dapat menggunakan aplikasi *standalone* yang dibangun dalam beragam bahasa pemrograman seperti C, C++, Java, Pascal, VB dan sebagainya. Bahasa-bahasa pemrograman tersebut telah mendukung koneksi ke berbagai macam DBMS baik secara langsung maupun melalui *Open Database Connectivity* (ODBC). Dapat juga menggunakan fitur DBMS tempat *data warehouse* dikembangkan. Pada SQL Server 2000 terdapat fitur *Data Transformation Services* (DTS) yang dapat digunakan untuk melakukan impor dan ekspor data dari dan ke berbagai macam DBMS.

2. Analisis Kasus Data warehouse Departemen Pertanian

Pembuatan *data warehouse* Departemen Pertanian ini merupakan inisialisasi bagi pembangunan *data warehouse* Departemen Pertanian secara menyeluruh. Pembuatan *data warehouse* ini mengambil dua *domain* dari bisnis proses Departemen Pertanian yaitu Basis Data Statistik Pertaniandan Basis Data Ekspor Impor.

Pembuatan *data warehouse* ini memiliki tujuan untuk memenuhi kebutuhan analisis secara umum eksekutif Departemen Pertanian. Pembuatan *data warehouse* ini berusaha membangun sebuah sistem dinamis yang mudah dikembangkan apabila terdapat kebutuhan baru dari pihak eksekutif.

Selama ini, pihak eksekutif Departemen Pertanian melakukan akses terhadap data laporan melalui Pusat Data dan Informasi Departemen Pertanian (PUSDATIN). Walaupun memiliki hak akses langsung ke data di masing-masing sistem operasional di bawahnya, tetapi data tersebut adalah data transaksional yang terlalu detail dan kurang bisa dijadikan acuan analisis.

7.1. Kebutuhan Sistem Data warehouse Departemen Pertanian

Pelaksanaan proses, ekstraksi data operasional kemudian menampilkannya dalam bentuk laporan, secara manual mempunyai banyak kendala. Perangkuman data yang melibatkan banyak sumber merupakan sebuah proses rumit yang rawan akan kesalahan. Tingkat kerumitan bisa bertambah apabila proses melibatkan banyak data yang redundan.

Secara umum, kelemahan-kelemahan dari sistem yang sekarang ada adalah :

1. Akses
   Pihak eksekutif harus melalui pihak ketiga, dalam hal ini pihak PUSDATIN, untuk mengakses laporan-laporan yang dibutuhkannya. Walaupun pihak eksekutif memiliki akses langsung ke sistem operasional, data pada sistem operasional kurang bisa dijadikan acuan untuk melakukan analisis.

2. Waktu
   Setelah terjadi sebuah transaksi pada sistem operasional, pihak eksekutif tidak bias mendapatkan laporannya dalam waktu yang cepat. Pihak eksekutif harus melewati proses birokrasi yang panjang untuk mendapatkannya.

3. Format
   Format laporan biasanya didefinisikan oleh pihak eksekutif pada awal permintaan laporan. Setelah laporan dengan format tertentu bisa diaksesnya, pihak eksekutif harus melakukan proses yang sama untuk mendapatkan informasi dalam format yang berbeda. Intinya adalah pihak eksekutif memiliki kesulitan untuk memandang informasi yang sama dalam bentuk yang berbeda, misalnya *spreadsheets* atau *chart*.

4. Integritas
   Walaupun informasi yang dibutuhkan telah didapatkan, keakuratan dari data masih diragukan. Proses manual yang melibatkan data besar dan rendundan rawan akan kesalahan.

Data warehouse menyediakan kemampuan untuk menanggulangi kelemahan-kelemahan diatas. Dengan *data warehouse*, pihak eksekutif memiliki hak penuh untuk mendapatkan laporan yang dibutuhkannya. Waktu yang dibutuhkan untuk mendapatkan laporan yang dibutuhkannya menjadi singkat. Dengan bantuan aplikasi pelaporan yang mendukung *data warehouse*, pihak eksekutif bisa melihat informasi yang dimilikinya dalam berbagai format.

Proses otomasi dalam *data warehouse*, meminimalisir kesalahan dan membuat keakuratan data lebih terjamin.

7.2. Penelitian Umum Data warehouse Departemen Pertanian

Seperti telah disebutkan sebelumnya, pembuatan *data warehouse* ini merupakan inisialisasi dibangunnya *data warehouse* keseluruhan Departemen Pertanian. Sebagai awalnya, *data warehouse* Departemen Pertanian akan melibatkan data dari tiga domain yaitu produksi, keuangan dan SDM. *Data warehouse* Departemen Pertanian dibangun dengan tujuan memberikan kemudahan pengaksesan informasi terhadap data penting dalam bisnis proses Departemen Pertanian yang dilakukan oleh pihak eksekutif Departemen Pertanian yang dibutuhkan dalam pengambilan keputusan strategis Departemen Pertanian. Sistem ini mengambil data dari tiga domain sumber, mengintegrasikanya dalam basis data tunggal dan menampilkannya kepada pengguna dalam bentuk laporan-laporan.

Data berasal dari sistem operasional dengan berbagai format basis data sumber dan format data. Sistem operasional ada yang menggunakan *Database Management System* (DBMS), ada juga yang menggunakan file *spreadsheet* (Microsoft Excel). Proses ekstraksi dijadwalkan oleh administrator *data warehouse*. Pendefinisian jadwal ini telah dilakukan sebelumnya oleh pihak eksekutif, pihak sistem operasional dan administrator *data warehouse*. Selain melakukan penjadwalan ekstraksi, administrator sebenarnya juga melaksanakan tugas standar dalam pemeliharaan basis data *data warehouse*. Di antaranya adalah proses *backup*, *recovery*, dan pengeksekusian proses ekstraksi (di luar jadwal) apabila terdapat kegagalan roses ekstraksi.

Tugas standar administrator ini tidak digambarkan pada Gambar III-5. Pengguna yang ingin melihat laporan akan berhadapan dengan antarmuka *Online Analytical Processing* (OLAP) t*ools*, yang akan menyampaikan informasi dalam *data warehouse* dengan berbagai format.

Proses ekstraksi dijadwalkan secara *default* sebulan sekali untuk keseluruhan data system operasional. Apabila terdapat perubahan kebutuhan, administrator dapat mengganti jadwal ekstraksi sesuai kebutuhan.

Hanya administrator yang memiliki hak akses langsung ke basis data *data warehouse*.

Pengguna tidak memiliki hak untuk mengakses langsung data pada basis data *data warehouse*. Antar muka pengguna hanyalah OLAP *tools*, seperti telah disebutkan di atas.

Basis data pada *data warehouse* Departemen Pertanian dibangun di atas DBMS SQL Server 2000. Tipe-tipe penyimpanan data menggunakan tipe standar yang disediakan DBMS seperti *char*, *varchar*, *number*, serta *datetime*. Fitur-fitur DBMS yang digunakan adalah fitur standar pendefinisian basis data dan tabel serta *foreign key*.

Untuk proses ETL digunakan fitur *Data Transformation Service* (DTS) menggunakan *ActiveX Script Task* dalam bahasa VBScript.

Data sumber akan diekstrak, ditransformasi, dan di-*load* ke *integration layer*. Kemudian data akan diekstrak lagi menjadi beberapa *data mart*. Aplikasi *report* akan berhubungan dengan *data mart* yang mempunyai data yang dibutuhkannya.

Penulisan Penelitian ini hanya melingkupi proses pada *flow* 1 (ETL dari data sumber ke *integration layer*). Untuk proses ekstrak data ke *data mart* dan proses penggunaan data pada *data mart* tidak dibahas dalam Penelitian ini.

Setelah sistem *data warehouse* mendapat data yang telah diekstrak dari sistem operasional, sistem akan melakukan proses transformasi sederhana yang termasuk di dalamnya proses *cleaning* sederhana dan integrasi sederhana. Sistem memiliki *dictionary* tentang nilai data yang mungkin terjadi dan data koreksi bila terdapat kesalahan nilai data. Proses integrasi juga didefinisikan dalam *dictionary* berbentuk keterangan nilai *reference* terhadap data lain apabila ada. Contoh proses *cleaning* adalah mengubah nilai data “palawija” menjadi “non palawija”.

2.3. Analisis Sistem Data warehouse Departemen Pertanian

Agar dapat dikembangkan menjadi *data warehouse* Departemen Pertanian yang terintegrasi secara keseluruhan, sistem inisialisasi ini berusaha dikembangkan dengan memikirkan perkembangan kebutuhan yang dinamis. Perubahan-perubahan kebutuhan berusaha diakomodasi oleh sistem tanpa perlu mengubah struktur sistem ataupun program-program yang digunakan sistem. Perubahan kebutuhan diharapkan hanya bersifat mengubah konfigurasi sistem.

Sistem yang dibangun ini memiliki kelemahan dalam proses integrasi. Proses integrasi kurang dilakukan menyeluruh terhadap data dalam seluruh sistem. Selain disebabkan oleh data sumber yang tidak menyeluruh (hanya sebagian sistem operasional), hal ini disebabkan karena kotornya data pada sistem operasional yang menyebabkan diperlukan proses integrasi lanjut yang lebih rumit. Sebenarnya ada cara yang lebih mudah yaitu mengubahproses bisnis sistem operasional, tetapi hal ini cenderung tidak dipilih karena membuat *data warehouse* bergantung terhadap sistem operasional.

Terdapat juga suatu informasi yang sama, tetapi disimpan sebagai subjek yang berbeda dalam sistem ini. Pada intinya proses integrasi lanjut dapat dikembangkan pada sistem ini tanpa mempengaruhi keseluruhan sistem.

2.4. Analisis Proses ETL Sistem Data warehouse Departemen Pertanian

Dalam rangka mendukung dibangunnya sebuah sistem *data warehouse* yang dinamis perlu dibangun proses ETL yang dinamis pula. Proses ETL bergantung pada skema data sumber, proses transformasi, dan skema data output (basis data *integration layer*).

Proses ETL *data warehouse* Departemen Pertanian dilakukan langsung *on the fly* tanpa melibatkan table sementara sebagai perantara. Data diekstrak kemudian dilakukan transformasi dan akhirnya di-*load* ke basis data secara langsung. Proses ini menghasilkan waktu proses ETL yang lebih cepat tetapi di sisi lain menghabiskan *resource* pada sistem operasional dan sistem *data warehouse* sendiri. Untuk itu penjadwalan proses ETL dilakukan ketika sistem operasional sedang tidak sibuk melakukan transaksi. Seperti telah disebutkan sebelumnya, proses transformasi hanya melakukan proses sederhana yaitu pengoreksian nilai data menggunakan tabel d*ictionary*. Setelah ditransformasi, data langsung di-*load* ke basis data *data warehouse*.

3. Kesimpulan Penerapan

Dalam penerapan ini telah dipilih beberapa metode yang digunakan untuk merealisasikan pembangunan proses ETL dinamis. Digunakan metode *file compare* untuk melakukan ekstraksi data sumber. Fungsi-fungsi yang digunakan dalam proses transformasi adalah konversi format, manipulasi string, fungsi aritmatik, *conditional assignment*, *referential integrity* dan *surrogate key resolution*. Proses *loading* dibagi menjadi dua yaitu *loading* untuk tabel dimensi dan *loading* untuk tabel fakta. Untuk *loading* tabel dimensi, dipilih strategi penggunaan *critical column* disertai dengan penambahan *record*.

I. Daftar Pustaka

1. Fathansyah, Ir., 2002, *Buku Teks Ilmu Komputer Basis Data*, Informatika, Bandung
2. Fowler, Martin, 2004, *UML Distilled Edisi 3 Panduan Singkat Bahasa Pemodelan Objek Standar*, 2005, Andi, Yogyakarta
3. http://en.wikipedia.org/wiki/Extract,_transform,_load
4. http://www.cert.or.id/~budi/courses/ec7010/dikmenjur-2004/supawi-report.pdf
5. http://www.ganesha.co.id
6. http://www.itech.fgcu.edu/cis/slides/chapter10.ppt
7. http://www.iwaysoftware.com/products/images/etl_chart_sm4.gif

8. http://www.mcrit.com/ASSEMBLING/assemb_central/WhatESS.htmhttp://www.ptct.com/EIS.html
9. http://www.utminers.utep.edu/mmahmood/cis5311dtmba/slides/chapter02.ppt
10. Ibrahim, Nugroho Setyabudhi, Takariyana Heni A., 2004, *Perancangan Data Warehouse Pada Pusat Data dan Informasi Pertanian*, Tesis Magister Manajemen Informasi Universitas Bina Nusantara, Jakarta
11. Inmon, W.H., 2002, *Building The Data Warehouse*, Third Edition, John Wiley and Sons, Inc., New York
12. O'Neil, Bonnie, Michael Schrader, John Dakin, Kieron Hardy, Matthew Townsend, Michael Whitmer, 1997, *Oracle® Data Warehousing*, Norhern Lights Software, Ltd., SAMS Publishing
13. Pressman, Roger S., Ph.D., 2002, *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku II)*, Andi, Yogyakarta
14. Suhendar, A. S.Si., Hariman Gunadi, S.Si., MT., 2002, *Visual Modeling Menggunakan UML dan Rational Rose*, Informatika, Bandung
15. Turban, Efram, Jay E. Aronson, and Ting Peng Liang, 2005, *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*, Edisi 7 Jilid 1, Andi, Yogyakarta
16. www.athens.edu/dreyfp/MIS/Third%20Ed/PPP/Ch10.ppt
17. www.itee.adfa.edu.au/courses/ACSC7309/work/ch08.ppt
18. www.vancouver.wsu.edu/fac/roseg/MIS350/ch2.ppt