DESAIN ETL DENGAN CONTOH KASUS PERGURUAN TINGGI

Spits Warnars
Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Petukangan Selatan, Kebayoran Lama, Jakarta 12260, Telepon: 021-5853753
E-mail: spits@bl.ac.id

ABSTRAK: Data Warehouse untuk perguruan tingi merupakan sebuah paradigma untuk membantu manajemen tingkat atas dalam rangka membuat keputusan yang efektif, efisien dan strategis berdasarkan laporan-laporan yang dapat dipercaya dan dihandalkan yang dihasilkan oleh Data Warehouse. Data Warehouse bukanlah sebuah perangkat lunak, perangkat keras atau perkakas akan tetapi Data Warehouse adalah suatu keadaan dimana database transaksi dimodelkan pada model lain yang ditujukan untuk membuat keputusan. *Extraction, Transformation and Loading* (ETL) adalah sebuah jembatan untuk membentuk Data Warehouse dan merubah data dari basisdata transaksi. Didalam setiap tabel fakta dan dimensi akan diselipkan dengan fields yang mencirikan pemanggilan *constructive merge* sebagai sebuah ETL proses. Proses ETL membutuhkan tabel ETL dan proses ETL dimana tabel ETL berfungsi sebagai padanan tabel antara tabel basisdata OLTP dan tabel pada Data Warehouse dan proses ETL akan mentransformasikan data yang berasal dari tabel pada basisdata OLTP kedalam tabel pada Data Warehouse berdasarkan tabel ETL. Proses ekstraksi akan dijalankan dengan sebuah tabel basisdata yang menggambarkan proses ETL dan sebuah algoritma ETL yang akan dijalankan secara otomatis pada saat proses transaksi tidak dijalankan bersamaan pada saat pelaksanaan backup transaksi basisdata harian.

Kata kunci: Data *warehouse*, ETL, algoritma ETL, Tabel ETL, pemanggilan *constructive merge*.

ABSTRACT: *Data Warehouse for higher education as a paradigm for helping high management in order to make an effective and efficient strategic decisions based on reliable and trusted reports which is produced from Data Warehouse itself. Data Warehouse is not a software, hardware or tool but Data Warehouse is an environment where the transactional database is modelled in other view for decision making purposes. ETL (Extraction, Transformation and Loading) is a bridge to build Data Warehouse and transform data from transactional database. In every fact and dimension table will be inserted with fields which represent the construction merge loading as an ETL (Extraction, Transformation and Loading) extraction. ETL needs an ETL table and ETL process where ETL table as table connectivity between tables in OLTP database and tables in Data Warehouse and ETL process will transform data from table in OLTP database into Data Warehouse table based on ETL table. The extraction process will be run with a table database as differentiate ETL process and an ETL algorithm which will be run automatically in idle transactional process, along with daily transactional database backup when the information system are not used.*

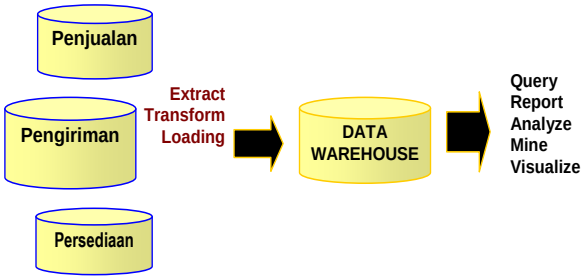
Keywords*: Data warehouse, ETL, ETL algorithm, ETL table, constructive merge loading.*

PENDAHULUAN

ETL merupakan sebuah proses penting yang merupakan bagian dari Data Warehouse yang berfungsi untuk mentransfer data yang ada pada basisdata transaksi kedalam basisdata Data Warehouse yang sudah dimodelkan sedemikian rupa berdasarkan *hypercubes* yang dibentuk berdasarkan laporan-laporan yang sering dipakai manajemen tingkat atas. Perguruan tinggi. Basisdata transaksi perguruan tinggi dipakai sebagai sebuah contoh. ETL dapat dikembangkan sesuai dengan pendekatan masing-masing organisasi didalam menjalan Data Warehouse tersebut, bisa jadi algoritma ETL yang dipakai akan berbeda disesuaikan dengan pilhan teknologi yang digunakan dan pendekatan Data Warehouse yang digunakan[1, 2, 3, 4]. ETL diaplikasikan dengan perangkat lunak yang bertanggung jawab untuk mengekstrasi data dari basisdata transaksi[6] dan yang siap dipakai sebagai sebuah aplikasi seperti ARKTOS[8].

Proses ETL dapat dimodelkan dengan gambar notasi yang lebih dimengerti dan dapat dikombinasikan dengan menggunakan desain berorientasi objek *Unified Modeling Language* (UML) [6] atau framework bentukan [9]. Model Manajemen ETL dapat diskenariokan dengan operator bentukan seperti *Search, Match, Create, Import, Merge, Rewrite, Invert,* dan *Deploy* [5] ataupun dengan bentukan operator lainnya [7].

Pementasan data sebagai bagian proses dari ETL menyediakan sebuah tempat dan area dengan sekumpulan fungsi untuk membersihkan, merubah, mengkombinasikan, merubah, penggandaan, dan menyiapkan sumber data untuk penyimpanan dan digunakan pada *Data Warehouse*. Ada 3 proses pementasan Data yang harus dilakukan oleh ETL [1].



Gambar 1. Lingkungan *Data Warehouse* dan lingkungan analisa

Gambar 1 memperlihatkan bagaimana konsep ETL dilakukan dan berhubungan dengan sistem aplikasi *Online Transactional Processing* (OLTP) seperti Penjualan, Pengiriman atau Persediaan dan pada nantinya *Data Warehouse* digunakan untuk menampilkan laporan dalam bentuk *query*, laporan, *data mining* dan lain-lain. Tujuan dari Pementasan Data adalah mengumpulkan, menseleksi, mengolah dan menggabungkan data yang relevan dari berbagai sumber untuk disimpan dalam *Data Warehouse*. Masing-masing proses pementasan d*ata* tersebut adalah [1]:

1) *Ekstraksi*, Mengambil data dari sistem *OLTP*, Metode yang dipakai ada 3 yaitu:
   a. Statis, umumnya dilakukan pada saat pemuatan data awal dan dilakukan pada saat sistem *OLTP* tidak berjalan.
   b. Terjadwal, ada 2 model pengambilan data terjadwal yaitu:
      a) Berdasarkan waktu, setiap pengaksesan *record* basisdata *OLTP* ditandai dengan waktu berupa tanggal dan jam dan secara terjadwal sebuah aplikasi program akan dijalankan untuk mengakses *Data Warehouse* berdasarkan perubahan waktu pada basisdata *OLTP*.
      b) Berdasarkan perbedaan berkas lama dan baru, adanya *backup* harian terhadap basisdata *OLTP*, dan secara terjadwal sebuah aplikasi program akan dijalankan untuk mengakses *Data Warehouse* jika sebuah tabel hari ini dan duplikat tabel hari sebelumnya berbeda.
   c. Seketika, ada 3 model pengambilan data seketika yaitu:
      a) Dengan mekanisme log transaksi, setiap transaksi *insert,update* dan *delete* terhadap sebuah *record*, maka basisdata seketika akan menulis pada *log* berkas yang digunakan untuk memilih transaksi yang telah berhasil dilaksanakan (*Committ*). *Log* berkas tidak berlaku jika kita menggunakan sistem data yang menggunakan indeks atau berkas teks.
      b) Dengan mekanisme basis data *trigger*, sama seperti log transaksi pilihan ini hanya untuk sistem yang menggunakan aplikasi basisdata. Trigger adalah sebuah program kecil yang akan dijalankan ketika sebuah event yang telah didefinisikan terjadi.
      c) Dengan sumber aplikasi, modifikasi program aplikasi *OLTP* agar secara seketika mengakses *Data Warehouse* setiap ada proses *insert,update* dan *delete* pada sebuah *record*.
2) *Transformation*, adalah proses pengambilan data mentah yang belum bisa disimpan pada *Data Warehouse*, oleh karena itu data harus sesuai standar struktur *Data Warehouse* yang telah ditentukan sehingga bisa disimpan ke *Data Warehouse*. *Transformation* data terdiri dari beberapa tahap yaitu:
   a. Seleksi, men-*select record* dari tabel basisdata *OLTP*, tahap ini merupakan bagian dari proses pengambilan data.
   b. Pemisahan dan Penggabungan, manipulasi data yang dibutuhkan untuk men-*select record OLTP*, yaitu melakukan proses pemisahan dan penggabungan bila dibutuhkan.
   c. Konversi, dilakukan untuk 2 alasan yaitu:
      a) Standarisasi pengambilan data dari berbagai sumber
      b) Membuat *field* dapat digunakan *Data Warehouse* dan dipahami oleh pengguna
   d. Ringkasan, data yang terlalu detail tidak selalu dibutuhkan pada *Data Warehouse* oleh karena itu perlu diringkas berdasarkan kebutuhan *Data Warehouse*.
   e. Pengayaan, menggunakan sebuah atau beberapa *field* untuk membuat hasil data yang terbaik pada *Data Warehouse*, prinsip ini merupakan pengembangan dari kumpulan sebuah atau beberapa *field* dari beragam *record* yang menghasilkan sebuah *field* untuk *Data Warehouse*.

   Dalam mentransformasikan data ada beberapa tipe fungsi transformasi yaitu:
   a. Revisi Format, merubah tipe data dan panjang *field* agar dapat distandarkan dan disesuaikan

dengan standar struktur penyimpanan pada *Data Warehouse*.

b. Penghilangan pengkodean *field, p*engkodean harus dihilangkan agar mudah dimengerti pengguna. Contoh: kode 1, 2 untuk jenis kelamin harus dirubah menjadi Pria, Wanita.
c. Menghitung dan menghasilkan nilai, Menghitung dan menghasilkan nilai rata-rata yang dibutuhkan.
d. Memecah *field,* dibutuhkan agar *Data Warehouse* mudah dipelihara dan dimengerti pengguna. Contoh: Nama harus dipecah menjadi nama depan, nama panggilan, nama keluarga.
e. Penggabungan informasi, bukan kebalikan dari memecah *field*, tapi menggabungkan data dari bermacam-macam sumber data.
f. Konversi kumpulan karakter, jika data berasal dari sumber yang berbeda arsitekturnya baik dari perangkat keras atau sistem operasi yang berbeda. Contoh: konversi dari EBCDIC ke ASCII.
g. Konversi unit pengukuran, digunakan untuk perusahaan yang mempunyai cabang di luar negeri.
h. Konversi tanggal dan jam, menggunakan format tanggal dan jam yang satu dan standar.
i. Ikhtisar, menjumlahkan transaksi
j. Restrukturisasi kunci, hindarkan pembuatan kunci yang mempunyai arti membingungkan. Ubahlah beberapa kunci ke kunci yang bersifat umum.
k. Deduplikasi, menghilangkan duplikasi data yang berasal dari sistem yang berjalan.

3) *Loading, a*da 4 mode *Loading* yaitu:
   a. Panggil, jika data sudah ada pada tabel *Data Warehouse* maka proses panggil ini akan menghapus data yang sudah ada dan menggantinya, jika data belum ada maka proses ini akan mengisi tabel *Data Warehouse*.
   b. Tambah, jika data sudah ada pada tabel *Data Warehouse* maka proses tambah ini akan menambah data dan ada kemungkinan terdapat duplikat *record* dan jika dikehendaki dimungkinkan duplikat *record* ditolak.
   c. *Destructive Merge,* jika kunci *record* yang datang cocok dengan kunci *record* yang ada maka akan merubah *record* yang ada dan jika *record* yang datang adalah *record* baru maka akan menambah *record* baru.
   d. *Constructive Merge,* jika kunci *record* yang datang cocok dengan kunci *record* yang ada maka akan menambah *record* baru dan menandai *record* baru tersebut sebagai penghubung ke *record* lama yang cocok dengan *record* yang akan datang.

TEKNIS DARI OLTP KE DATA WAREHOUSE

Dibandingkan dengan basisdata transaksional OLTP, Data Warehouse mempunyai struktur yang tidak normal. Ada 2 jenis tabel yang terdapat pada Data Warehouse yaitu:

1) Tabel dimensional, berisi penjelasan detail dari setiap informasi yang didapatkan dari hasil analisa kebutuhan sistem, sebuah field yang berlaku sebagai *primary key* akan terhubung ke tabel fakta dan berlaku sebagai *foreign key* pada tabel fakta.
2) Tabel fakta, berisi data rinci dan agregat yang merupakan kumpulan dari beberapa atribut yang bersifat sebagai *Foreign Key* sebagai penghubung ke tabel dimensional, dan atribut tambahan lainnya yang merupakan nilai data. Pada tabel ini primary key merupakan *composite/compound key* (*Primary key* yang merupakan gabungan dari beberapa *foreign key*).

Mendesain model *Data Warehouse* dapat digunakan konsep dimensi bisnis yang merupakan dasar untuk mendefinisikan kebutuhan *Data Warehouse*. Dimensi bisnis ini merupakan pandangan manajer bisnis terhadap sistem yang berjalan sesuai dengan kebutuhan manajemen tingkat atas sebagai pengguna *Data Warehous*e [1]. Dimensi bisnis biasa digambarkan dengan kubus, dengan asumsi dimensi yang ada pada sebuah *Data Warehouse* umumnya mencapai 3 dimensi. Jika dimensi bisnis lebih dari 3 dimensi maka disebut bermacam dimensi yang menampilkan kubus multidimensi yang disebut juga *hypercube.* Setiap *hypercube* yang terbentuk dari setiap laporan merupakan komponen rangkaian kecil yang jika dihubungkan dengan *hypercube* lainnya akan membentuk sebuah desain model *Data Warehouse* yang juga merupakan *hypercube*. *Hypercube* ini merupakan sebuah alat untuk menangkap kebutuhan user dan menggambarkan model dimensi bisnis atau struktur basisdata Data Warehouse.

Membentuk Data Warehouse langkah pertama yang harus dilakukan adalah membentuk tabel dimensional. Ada 3 Pendekatan yang dipakai dalam pengembangan model dimensi [1], adapun tipe pendekatan tersebut adalah:

1) Mengacu pada data, pengembangan model dimensi yang berorientasi ke sistem *OLTP*.
2) Mengacu pada pengguna, pengembangan model dimensi yang berorientasi kepada kebutuhan pengguna yang dilakukan dengan menggunakan

metode pengumpulan data seperti wawancara, *brainstorming*, *Joint Application Development* (JAD), dan lain-lain.

3) Mengacu pada tujuan, pengembangan model dimensi yang berorientasi kepada tujuan organisasi yang tercantum dalam visi dan misi perusahaan.

Pada tulisan ini pendekatan yang dipakai dalam pengembangan model dimensi adalah mengacu pada pengguna dengan pertimbangan sebagai berikut:

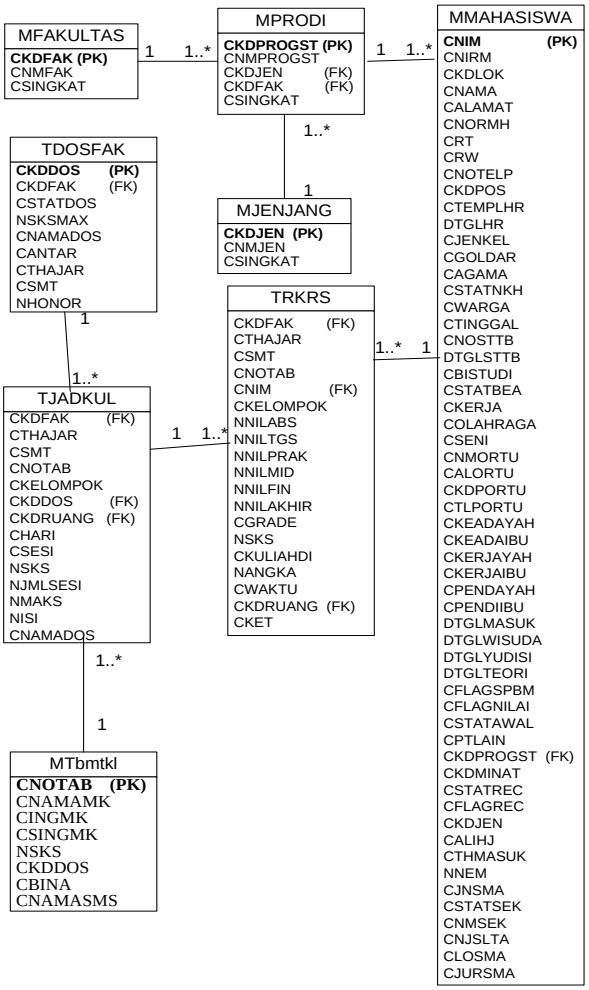
1) Memuaskan kebutuhan berbagai tingkat manajemen, terutama manajemen tingkat menengah sampai manajemen tingkat atas.
2) Mendapatkan kebutuhan laporan dan mendesain *Data Warehouse* yang memang dibutuhkan oleh manajemen.

Oleh karena pengembangan model dimensi untuk pembentukan model data warehouse menggunakan tipe pendekatan yang mengacu pada pengguna, maka pembentukan dimensi akan dibentuk berdasarkan laporan-laporan yang dihasilkan oleh sistem OLTP dan yang sering dipakai oleh manajemen terutama manajemen tingkat atas sebagai dasar bahwa laporan-laporan berorientasi sebagai laporan-laporan yang memuaskan kebutuhan pengguna. Sebagai contoh akan dibatasi pada 6 buah laporan-laporan yang sering digunakan atau dibutuhkan oleh manajemen tingkat atas perguruan tinggi. Adapun laporan-laporan tersebut adalah:

1) Laporan Jumlah Mahasiswa Per Jenjang Program Studi Per Jenis Kelamin Per Angkatan.
2) Laporan Jumlah Mahasiswa Aktif Per Semester Tahun Ajaran Per Jenjang Program Studi Per Jenis Kelamin Per Angkatan.
3) Laporan Jumlah Komposisi Indeks Prestasi Per Semester Tahun Ajaran Per Jenjang Program Studi Per Angkatan
4) Laporan Jumlah Komposisi Grade Nilai Per Semester Tahun Ajaran Per Jenjang Program Studi Per Jenis Kelamin Per Angkatan
5) Laporan Pengajaran Dosen Per Semester Tahun Ajaran
6) Laporan Pengajaran Mata Kuliah Oleh Dosen Per Semester Tahun Ajaran

Keenam laporan tersebut, masing-masing dibentuk berdasarkan basisdata OLTP yang terdapat pada Gambar 2.

Untuk menghasilkan laporan Jumlah Mahasiswa Per Jenjang Program Studi Per Jenis Kelamin Per Angkatan digunakan 4 tabel basisdata OLTP yang terdapat pada Gambar 2 yaitu tabel Mmahasiswa, Mprodi, Mfakultas dan Mjenjang. Laporan ini membentuk *hypercube* dimana *hypercube* laporan ini membentuk tabel fakta Wdata1 dan tabel dimensi Wprodi yang terdapat pada Gambar 3 dan dengan Data Warehouse maka laporan ini akan dihasilkan dengan menggunakan 1 tabel fakta Wdata1 dan 1 tabel dimensi Wprodi.



Gambar 2. *Class Diagram* model data logika *OLTP*

Untuk menghasilkan Laporan Jumlah Mahasiswa Aktif Per Semester Tahun Ajaran Per Jenjang Program Studi Per Jenis Kelamin Per Angkatan digunakan 5 tabel basisdata OLTP yang terdapat pada Gambar 2 yaitu tabel Mmahasiswa, Trksrs, Mprodi, Mfakultas dan Mjenjang. Laporan ini membentuk *hypercube* dimana *hypercube* laporan ini membentuk tabel fakta Waktif dan tabel dimensi Wprodi yang terdapat pada Gambar 3 dan dengan Data Warehouse maka laporan ini akan dihasilkan dengan menggunakan 1 tabel fakta Waktif dan 1 tabel dimensi Wprodi.

Untuk menghasilkan Laporan Jumlah Komposisi Indeks Prestasi Per Semester Tahun Ajaran Per Jenjang Program Studi Per Angkatan digunakan 4 tabel basisdata OLTP yang terdapat pada Gambar 2 yaitu tabel Trksrs, Mprodi, Mfakultas dan Mjenjang.

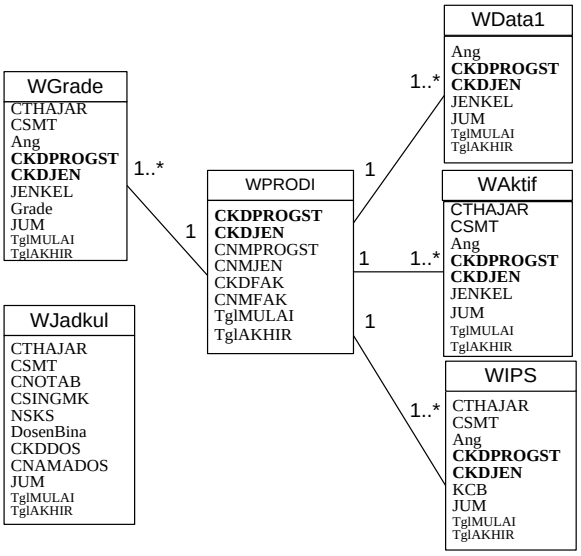
Laporan ini membentuk *hypercube* dimana *hypercube* laporan ini membentuk tabel fakta Wips dan tabel dimensi Wprodi yang terdapat pada Gambar 3 dan dengan Data Warehouse maka laporan ini akan dihasilkan dengan menggunakan 1 tabel fakta Wips dan 1 tabel dimensi Wprodi.

Untuk menghasilkan Laporan Jumlah Komposisi Grade Nilai Per Semester Tahun Ajaran Per Jenjang Program Studi Per Jenis Kelamin Per Angkatan digunakan 5 tabel basisdata OLTP yang terdapat pada Gambar 2 yaitu tabel Mmahasiswa, Trksrs, Mprodi, Mfakultas dan Mjenjang. Laporan ini membentuk *hypercube* dimana *hypercube* laporan ini membentuk tabel fakta Wgrade dan tabel dimensi Wprodi yang terdapat pada Gambar 3 dan dengan Data Warehouse maka laporan ini akan dihasilkan dengan menggunakan 1 tabel fakta Wgrade dan 1 tabel dimensi Wprodi.

Untuk menghasilkan Laporan Pengajaran Dosen Per Semester Tahun Ajaran digunakan 4 tabel basisdata OLTP yang terdapat pada Gambar 2 yaitu tabel Tjadkul, Tdosfak, Mtbmtkl dan Mfakultas. Laporan ini membentuk hypercube dimana *hypercube* laporan ini membentuk tabel fakta Wjadkul dan tanpa tabel dimensi yang terdapat pada Gambar 3 dan dengan Data Warehouse maka laporan ini akan dihasilkan dengan menggunakan 1 tabel fakta Wjadkul.

Untuk menghasilkan Laporan Pengajaran Mata Kuliah Oleh Dosen Per Semester Tahun Ajaran digunakan 4 tabel basisdata OLTP yang terdapat pada Gambar 2 yaitu tabel Tjadkul, Tdosfak, Mtbmtkl dan Mfakultas sama dengan pembuatan laporan Pengajaran Dosen Per Semester Tahun Ajaran. Laporan ini membentuk *hypercube* dimana *hypercube* laporan ini membentuk tabel fakta Wjadkul dan tanpa tabel dimensi yang terdapat pada Gambar 3 dan dengan Data Warehouse maka laporan ini akan dihasilkan dengan menggunakan 1 tabel fakta Wjadkul sama dengan pembuatan laporan Pengajaran Dosen Per Semester Tahun Ajaran.

Gambar 3 merupakan *class diagram* model data logika Data Warehouse yang merupakan penggabungan tabel fakta dan tabel dimensi yang dihasilkan oleh *hypercube* dari masing-masing laporan. Pada masing-masing tabel Data Warehouse ini ditambahkan *field* TglMULAI dan TglAkhir yang berfungsi sebagai proses pembaharuan data *record*, oleh karena pada *Data Warehouse* tidak boleh dilakukan proses penghapusan record. Data record yang masih berlaku adalah apabila *field* TglAkhir masih kosong, apabila *field* TglAkhir telah terisi maka akan terbentuk *record* duplikat yang menggambarkan data *record* yang terkini.



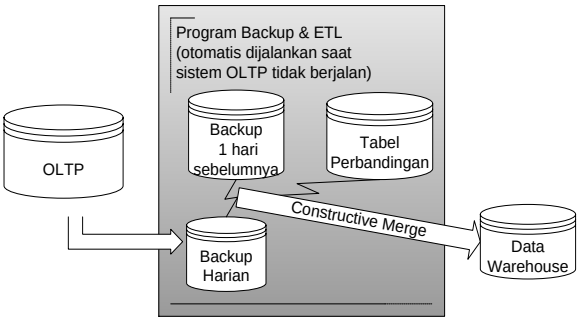
Gambar 3. *Class Diagram* model data logika *Data Warehouse*

Untuk mengambil data dari *OLTP* dipakai metode ekstraksi data terjadwal berdasarkan perbedaan berkas lama dan baru. Sesuai dengan penjelasan teori sebelumnya dengan adanya *backup* harian terhadap basis data *OLTP*, dan secara terjadwal sebuah aplikasi program akan dijalankan untuk mengakses *Data Warehouse* jika sebuah tabel hari ini dan duplikat tabel hari sebelumnya berbeda. Metode ini dipilih dengan alasan disesuaikan dengan kegunaannya bahwa perlu dilakukan *backup* data harian terhadapat basis data OLTP untuk menjaga keamanan data. Selanjutnya *backup* berkas hari ini dibandingkan dengan *backup* berkas satu hari sebelumnya, dengan asumsi proses ETL tidak perlu dilakukan setiap hari dimana proses ETL hanya akan dilakukan jika ada perbedaan data tabel harian antara tabel hari ini dan tabel hari sebelumnya.

Metode terjadwal berdasarkan perbedaan berkas lama dan baru yang dipilih untuk mentransformasi data pada basisdata OLTP kedalam Data Warehouse dengan alasan:

1) Tidak mengganggu dan memperlambat sistem *OLTP* dan ini terjadi jika model ekstrak seketika yang digunakan, dimana ekstrak dilakukan bersamaan pada saat sistem *OLTP* sedang berjalan.
2) Proses *backup* harian untuk menjaga konsistensi dan keamanan data sejalan dengan metode terjadwal berdasarkan perbedaan berkas lama dan baru ini. Dimana proses *backup* biasanya dilaksanakan pada saat sistem *OLTP* tidak berjalan.
3) Proses perbandingan menjadi lebih cepat dan hanya dilakukan terhadap *field* yang ditentukan pada tabel perbandingan ETL sebagai nama *field* OLTP.

Gambar 4 menggambarkan proses ETL yang dijalankan dalam mentransformasi data dari basisdata OLTP kedalam Data Warehouse. Program ETL ini akan dijalankan bersamaan dengan proses *backup* basisdata OLTP harian yang biasanya djalankan pada saat sistem OLTP tidak sedang digunakan yaitu pada saat selesainya transaksi harian dan akan dijalankan oleh sebuah program *server* yang dijalankan otomatis pada jam-jam tertentu dimana sistem OLTP ini tidak digunakan dan bersamaan dengan proses *backup* data. Untuk menjalankan proses ETL ini selain sebuah program kecil yang berfungsi melakukan fungsi ETL yang bertugas mentransformasikan data dari basisdata OLTP kedalam Data Warehouse juga dibutuhkan sebuah tabel yang dinamakan tabel perbandingan yang berfungsi mempadankan antara tabel yang terdapat pada basisdata OLTP dan tabel yang terdapat pada Data Warehouse. Berdasarkan tabel perbandingan yang sekaligus dapat disebut juga sebagai tabel ETL ini, program ETL akan melakukan proses transformasi data dari basisdata OLTP ke Data Warehouse berdasarkan padanan tabel yang terdapat pada tabel ETL ini.



Gambar 4. Teknis ETL dari *OLTP* ke *Data Warehouse*

Secara detail metode ini dilakukan dengan langkah-langkah sebagai berikut dengan dibuatnya:

1) Sebuah tabel basis data perbandingan yang dapat disebut juga sebagai tabel ETL yang terlihat pada tabel 1 dimana *field* tabel ini berisi:
   a. Nama tabel *Data Warehouse*, merupakan tabel fakta dan tabel dimensi keseluruhan untuk *Data Warehouse*. Semua tabel baik tabel fakta atau tabel dimensi harus terdapat pada field ini, dengan tujuan mempermudah dan membantu proses ETL didalam mendapatkan data pada masing-masing tabel Data Warehouse tersebut yang ditransformasikan dari tabel pada database OLTP.
   b. Nama tabel *OLTP* yang merupakan sumber data *OLTP* untuk masing-masing tabel *Data Warehouse*.
   c. P*rimary key* OLTP merupakan kunci untuk mengakses masing-masing tabel *OLTP*.
   d. Nama *field OLTP* merupakan *field* yang akan dibandingkan antara berkas *backup* hari ini dan berkas *backup* 1 hari sebelumnya.

   Isi dari *field* tabel perbandingan ini berisi dengan hasil dari *hypercubes* dari laporan diatas, berkas basisdata yang membuat masing-masing *Data Warehouse*, *primary key* dan *field OLTP* yang diekstrak.

Tabel 1. Tabel perbandingan *ETL*

Nama Tabel DW	Nama Tabel OLTP	Primary Key OLTP	Nama Field OLTP
WDATA1	MMAHASISWA	CNIM	CJENKEL
WPRODI	MPRODI	CKDPROGST	
	MFAKULTAS	CKDFAK	
	MJENJANG	CKDJEN	
WAKTIF	TRKRS	CKDFAK	
		CTHAJAR	
		CSMT	
		CNOTAB	
		CNIM	
WIPS	TRKRS	CKDFAK	CGRADE
		CTHAJAR	NSKS
		CSMT	
		CNOTAB	
		CNIM	
WGRADE	TRKRS	CKDFAK	CGRADE
		CTHAJAR	
		CSMT	
		CNOTAB	
		CNIM	
WJADKUL	TJADKUL	CKDFAK	CKDDOS
		CTHAJAR	CNAMADOS
		CSMT	
		CNOTAB	
		CKELOMPOK	
	MTBMTKL	CNOTAB	
	TDOSFAK	CKDFAK	
		CKDDOS	

2) Sebuah program yang secara otomatis dijalankan pada jam-jam tertentu pada saat aplikasi *OLTP* tidak berjalan dan biasanya antara pukul 21.00 wib sampai 07.00 wib. Program ini mempunyai urutan kegiatan sebagai berikut:
   a. Melakukan proses *backup* harian keseluruhan basisdata.
   b. Berdasarkan tabel ETL perbandingan berkas *backup* basisdata hari ini dibandingkan secara per *record* dan per *field* yang ditentukan sesuai dengan nama *field* OLTP yang akan dibandingkan terhadap berkas *backup* 1 hari sebelumnya.
   c. Jika ada perbedaan *field* atau *record* pada perbandingan maka akan merubah data pada *Data Warehouse* dengan konsep *constructive merge* dimana berdasarkan kunci *record* tersebut dicari ke *Data Warehouse*, jika cocok maka akan menambah *record* baru pada *Data Warehouse* dan mengisi *field* TglAkhir pada *record* lama. Jika tidak cocok maka akan menambah *record* baru pada *Data Warehouse*.

Berikut ini merupakan penggalan algoritma untuk program tersebut:

```
Select * from tabel_perbandingan
Buat array hasil select
Selama masih ada isi array
   //cek jika ada perubahan isi
nama_field_OLTP dan jika ada
Select count(primary_key) from
nama_tabel_OLTP-copitoday A,
 nama_tabel_OLTP-copilastday B
   where A.primary_key=B.primary_key and
A.nama_field_OLTP<>B.nama_field_OLTP
   Jika ada perubahan
      Update data pada Nama_tabel_DW
   Akhir jika
   //cek untuk data baru
   Select primary_key from nama_tabel_OLTP-
copitoday
   where not in
   (Select primary_key from
nama_tabel_OLTP-copilastday)
   Jika ada perubahan
      Insert data baru pada Nama_tabel_DW
   Akhir jika
   //cek untuk data yang dihapus
   Select primary_key from nama_tabel_OLTP-
copilastday
   where not in
   (Select primary_key from nama_tabel_OLTP-
copitoday)
   Jika ada perubahan
      Update data pada Nama_tabel_DW
   Akhir jika
Akhir selama
```

PROSENTASE EFISIENSI

Kualitas multidimensional pada data warehouse dapat diuji dengan matrik yang diujikan pada tabel star dan skema pada data warehouse berdasarkan jumlah field, jumlah foreign key pada tabel-tabel tersebut [10]. Prosentase efisiensi akan membuktikan bahwa penggunaan data warehouse lebih efektif daripada basisdata OLTP dan prosentase efisiensi dibuktikan dengan rumus prosentase kenaikan (1), dimana data lama adalah database OLTP dan data baru adalah data warehouse. Prosentase efisiensi akan dibuktikan pada total panjang record, total jumlah record dan total keseluruhan byte, antara basisdata OLTP dan data warehouse.

(data lama – data baru)/data baru * 100 (1)

Tabel 2 merupakan besaran 8 tabel basisdata OLTP yang terdapat pada Gambar 2, yang mempunyai total panjang record 1099 byte, total jumlah record 131171 record dan total keseluruhan byte adalah 31303511 byte. Tabel 3 merupakan besaran 6 tabel data warehouse yang terdapat pada Gambar 3, yang mempunyai total panjang record 326 byte, total jumlah record 1138 record dan total keseluruhan byte adalah 71555 byte.

Tabel 2. Tabel besaran isi tabel database OLTP

Nama Tabel	Panjang record	Jumlah record	Total byte
MMAHASIWA	586 byte	42977 record	25 184 522 byte
MPRODI	48 byte	16 record	768 byte
MFAKULTAS	65 byte	7 record	455 byte
MJENJANG	24 byte	3 record	72 byte
TRKRS	68 byte	84774 record	5 764 632 byte
TJADKUL	88 byte	1988 record	174 944 byte
TDOSFAK	73 byte	386 record	28 178 byte
MTBMTKL	147 byte	1020 record	149 940 byte
Total	**1099 byte**	**131171 record**	**31303511 byte**

Tabel 3. Tabel besaran isi tabel data warehouse

Nama Tabel	Panjang record	Jumlah record	Total byte
WPRODI	35 byte	16 record	560 byte
WJADKUL	127 byte	303 record	38481 byte
WGRADE	44 byte	368 record	16192 byte
WDATA1	34 byte	279 record	9486 byte
WAKTIF	43 byte	74 record	3182 byte
WIPS	43 byte	98 record	4214 byte
Total	**326 byte**	**1138 record**	**71555 byte**

Dengan menggunakan rumus prosentase efisiensi (1) prosentase efisiensi untuk total panjang record adalah 237.12, dimana data lama adalah 1099 byte dan data baru adalah 326 byte, sehingga (1099-326)/326*100 = 237.12. Dengan menggunakan rumus prosentase efisiensi (1) prosentase efisiensi untuk total jumlah record adalah 11426.45, dimana data lama adalah 131171 record dan data baru adalah 1138 record, sehingga (131171-1138)/1138*100 = 11426.45. Dengan menggunakan rumus prosentase efisiensi (1) prosentase efisiensi untuk total keseluruhan keseluruhan byte adalah 43467.48, dimana data lama adalah 31303511 byte dan data baru adalah 71555 byte, sehingga (31303511-71555)/71555*100 = 43467.48. Pembuktian rumus membuktikan bahwa penggunaan data warehouse lebih efisien 237.12% untuk panjang record, lebih efisien 11,426.45% untuk jumlah record dan lebih efisien 43,467.48% untuk total keseluruhan byte.

Dari keseluruhan efisiensi kenaikan prosentase jika digabungkan akan menghasilkan rata-rata efisiensi kenaikan prosentase 42951,51%. Dibuktikan dengan menggunakan rumus prosentase efisiensi (1), dimana data lama atau basisdata OLTP adalah 31435781 dan data baru atau data warehouse adalah 73019, sehingga (31435781-73019)/73019*100 = 42951.51. Dimana 31435781 adalah penggabungan total panjang record, total jumlah record dan total keseluruhan byte pada basisdata OLTP, sehingga 1099 + 131171 + 31303511 = 31435781. Sedangkan 73019 adalah penggabungan total panjang record, total jumlah record dan total keseluruhan byte pada data warehouse, sehingga 326 + 1138 + 71555 = 73019. Jadi rata-rata efisiensi kenaikan prosentase 42951,51% menunjukkan bahwa penggunaan data warehouse lebih handal dan efisien dibandingkan penggunaan basisdata OLTP.

KESIMPULAN

ETL merupakan sebuah konsep yang dibutuhkan pada Data Warehouse yang berfungsi mentransformasikan data pada basisdata OLTP kedalam Data Warehouse. ETL didalam pelaksanaannya membutuhkan tabel ETL dan proses ETL, dimana tabel ETL berisi padanan tabel basisdata OLTP dan tabel Data Warehouse, yang berfungsi membantu proses ETL didalam mentransformasikan data pada basisdata OLTP kedalam Data Warehouse. Semua tabel yang terdapat pada Data Warehouse harus terdapat pada tabel ETL ini.

Pilihan penerapan algoritma ETL yang akan digunakan dapat disesuaikan dengan desain model Data Warehouse yang digunakan, teknologi manajemen basisdata dan aplikasi perangkat lunak yang digunakan yang akan lebih baik jika sama dengan aplikasi perangkat lunak untuk menampilkan laporan dan query dari Data Warehouse dan teknologi manajemen basisdata yang digunakan untuk menyimpan data pada Data Warehouse.

Algoritma ETL ini dapat dikembangkan disesuaikan dengan kondisi dan jenis model Data Warehouse yang digunakan dan penggunaan basisdata ETL tidak akan benyak mengalami perubahan sebagaimana apa adanya, dimana ada basisdata transaksi yang dimasukkan, basisdata Data Warehouse, dan *primary key* dari masing-masing tabel serta nama fields yang akan diekstrak dari basisdata transaksi.

Penggunaan algoritma ETL ini hanya dibataskan pada penyimpanan data yang bersifat manajemen basisdata dan untuk penggunaan pada penyimpanan data bentuk lainnya, dimana sifat ETL yang dapat mengekstrak data dari bermacam-macam simpanan data baik yang terstruktur maupun tidak terstruktur.

DAFTAR PUSTAKA

1. Ponniah, P., 2001, *Data Warehousing Fundamentals*, John Willey & Sons, Inc, New York, USA.
2. McGuire, M., Gangopadhya, A., Komlodi, A., and Swan, C., 2008, *A User–centered design for a spatial data warehouse for data exploration in environmental research*, Ecological Informatics, vol. 3, Issue: 4-5, pp. 273-285.
3. Bara, A., Lungu, I., Velicanu, M., and Botha, I., 2008, *Improving query performance in Virtual Data Warehouses*, WSEAS transactions in Information Science and Applications, Vol. 5, Issue. 5, pp. 632-641.
4. Simitsis, A., Vassiliadis, P., and Sellis, T., 2005, *Optimizing ETL Processes in Data Warehouses*, Proceedings of the 21st International Conference on Data Engineering, pp. 564-575.
5. Albrecht, A., and Naumann, F., 2008. *Managing ETL Processes*, In Proceedings of the International Workshop on New Trends in Information Integration, pp. 12-15, Auckland, New Zealand.
6. Vassiliadis, P., Simitsis, A., and Skiadopoulos, S., 2002, *Conceptual Modeling for ETL Processes*, In Proceedings of Data Warehousing and OLAP (DOLAP'2002) ACM 5th Intl Workshop in conjunction with CIKM'02, pp. 14-21, McLean, USA.
7. Thiele, M., Kiefer, T., and Lehner, W., 2009, *Cardinality Estimation in ETL Processes*, In Proceeding of the ACM twelfth international workshop on Data warehousing and OLAP, pp. 57-64, Hong Kong, China.
8. Vassiliadis, P., Vagena, Z., Karayannidis, N., and Sellis, T., 2001, *ARKTOS: Towards The Modeling, Design, Control and Execution of ETL Processes*, Information Systems, Vol. 26, no. 8, pp. 537-561.
9. Vassiliadis, P., Simitsis, A., Georgantas, P., and Terrovitis, M., 2003, *A Framework for The Design of ETL Scenarios*, In Proceedings of Caise, pp. 520-535.
10. Calero, C., Piatiini, M., Pascual, C., and Serrano, M.A., 2001, *Towards Data Warehouse Quality Metrics*. Proceedings of the 3rd Intl. Workshop on Design and Management of Data Warehouses (DMDW'2001). Interlaken. Switzerland, 39: 2-11.