

# PERANCANGAN DATA WAREHOUSE PENJUALAN UNTUK PERHITUNGAN LABA RUGI (Studi Kasus di Era 5000 Lembuswana Samarinda)

<sup>1)</sup> Awang Harsa Kridalaksana

<sup>2)</sup> Hamdani

<sup>3)</sup> Norma Yanti

<sup>1,2,3)</sup> Program Studi Ilmu Komputer FMIPA Universitas Mulawarman  
Email : awangkid@yahoo.com <sup>1)</sup>, iniemaildani@yahoo.com <sup>2)</sup>, yantie07@gmail.com <sup>3)</sup>

## ABSTRAK

Data *Warehouse* merupakan sistem *repository* dan merupakan suatu data yang besar dan mengandung informasi dan di digunakan untuk keperluan analisis dan laporan. Selain itu data *warehouse* juga digunakan untuk mendukung pengambilan keputusan untuk menentukan kebijakan dan strategi yang akan diambil untuk meningkatkan mutu penjualan dan pendapatan kedepannya. *Database OLTP (OnLine Transaction Processing)* yang digunakan untuk penelitian ini adalah *database* penjualan era5000 Lembuswana Samarinda. *Database OLTP* tersebut diekstrak dan dilanjutkan ke data *warehouse* untuk kemudian digunakan dalam pembuatan laporan.

Hasil dari penelitian ini adalah sebuah data *warehouse* lengkap dengan pelaporan informasi hasil laba yang diperoleh mulai tahun 2006 sampai dengan 2010. *Platform database* yang digunakan untuk *OLTP* dan data *warehouse* pada penelitian ini adalah Microsoft SQL Server 2005. Untuk proses *ETL (Extraction, Transformation, Loading)* dilakukan di *Integration Service*. Sedangkan untuk proses pembuatan laporan dilakukan di *Report Service* menggunakan sistem *OLAP* yang dapat diakses melalui *browser Internet Explorer*.

Digunakan *query* seleksi untuk proses pemilihan data operasional dan menyalurkan ketujuan yaitu data *warehouse*. Dengan bahasa pemrograman yang digunakan untuk membuat *user interface* adalah bahasa pemrograman ASP dot Net yang merupakan teknologi Microsoft yang di khususkan untuk pengembangan *web* dinamis. Aplikasi data *warehouse* hanya dapat ditampilkan menggunakan *browser Internet Explorer*.

**Kata Kunci :** *Data Warehouse, OLTP, Report Service.*

## PENDAHULUAN

*Data Warehouse* merupakan sebuah teknologi yang memungkinkan data dapat diakses dengan mudah dan efisien untuk mendukung pengambilan keputusan. Untuk sebuah supermarket, teknologi ini dapat digunakan untuk menentukan kebijakan dan strategi apa saja yang akan diambil kedepannya. Pengolahan informasi yang baik dapat memberikan kemudahan bagi para pengambil keputusan untuk menentukan keputusan atau kebijakan apa saja yang harus diambil untuk memajukan supermarket yang dikelolanya.

Era5000 sebagai salah satu supermarket yang ada di Samarinda Kalimantan timur, hendaknya memiliki sebuah sistem *Data Warehouse* yang mampu memenuhi kebutuhan para pengambil keputusan dalam meningkatkan penjualan untuk menghadapi persaingan.

Keberhasilan sebuah supermarket dapat dinilai dari kemampuannya memperoleh laba dan memperkecil kemungkinan memperoleh kerugian. Dengan adanya pemanfaatan teknologi informasi dapat digunakan untuk mendapatkan, mengolah dan mempublikasikan informasi untuk menunjang kegiatan pada sebuah supermarket di Era5000.

Proses pengolahan terhadap data penjualan ini, manajemen bisa mendapatkan informasi yang digunakan untuk keperluan manajemen supermarket seperti menentukan jumlah barang yang harus disiapkan di gudang. Penggunaan *data warehouse* yang mengolah data penjualan sehingga menghasilkan informasi yang bisa digunakan untuk membantu manajemen supermarket dalam menentukan laba dan rugi. Dalam pembuatan *Data Warehouse* penjualan dengan tabel fakta Penjualan dan beberapa dimensi, dari datawarehouse dihasilkan informasi penjualan, seperti informasi penjualan kasir perwaktu, informasi penjualan produk perwaktu, dan lain-lain. Pada aplikasi

OLAP, penerapan hirarki pada dimensi Waktu (Tahun, Quarter, Bulan, Minggu, dan Tanggal) pada Cube Penjualan semakin menambah informasi yang dapat digunakan untuk membantu manajemen menentukan pengambilan kebijakan dalam suatu periode.

Berdasarkan penelitian sebelumnya mengenai *Data Warehouse* penjualan mobil sistem ini dapat memproses data penjualan dengan mencoba menerapkan pergudangan data. Jurnal ini juga membahas bagaimana sistem dapat mengkatagorikan data menurut dimensi yakni item, waktu, dan lokasi. Sehingga dapat menghasilkan suatu analisis.

*Data Warehouse* adalah database yang saling bereaksi yang dapat digunakan untuk *query* dan analisis, bersifat orientasi subjek, terintegrasi, *time-variant*, tidak berubah, yang digunakan untuk membantu para pengambil keputusan. Membangun pada penelitian sebelumnya dengan judul “Aplikasi *Data Warehouse* Untuk Analisis Penjualan Mobil Berbasis Multidimensional Modeling (MDM) dan Star Schema Desain (studi kasus: PT. Asco Automotive)” penulis Nugroho (2008) di Jurnal Teknologi Informasi. Dapat diujikan untuk suatu laporan data yang berbasis multidimensional modeling dengan banyaknya integrasi data pada suatu system laporan penjualan.

## TINJAUAN PUSTAKA

### Pengertian *Data Warehouse*

Sketsa pembuka menunjukkan sebuah skenario dimana sebuah *Data Warehouse* dapat digunakan untuk mendukung pengambilan keputusan, menganalisis sejumlah besar data dari berbagai sumber untuk menyediakan hasil cepat untuk mendukung hasil kritis. Data untuk *Data Warehouse* berasal dari berbagai sumber daya eksternal dan internal dan berisikan dan diorganisasi dalam suatu cara yang konsisten dengan kebutuhan organisasi. *Data Warehouse* dapat digambarkan sebagai data berorientasi-subjek, terintegrasi, *time-variant*, koleksi data *non-volatile* yang mendukung pengambilan keputusan analitik. *Data Warehouse* adalah kumpulan data dari berbagai sumber yang ditempatkan menjadi satu dalam dalam tempat penyimpanan berukuran besar lalu diproses menjadi bentuk penyimpanan multi-dimensional dan didesain untuk *querying* dan *reporting* (Turban,2005).

Pengetian *data warehouse* menurut Vincent Rainardi yang menulis buku “*Building a Data Warehouse With Examples in SQL Server*” beliau menyampaikan definisi sebagai berikut ;

*Data Warehouse* adalah sistem yang mengambil dan mengkonsolidasikan data *periodically* dari sistem sumber ke dalam sebuah menyimpan data dimensi atau dinormalisasi. biasanya menyimpan tahun sejarah dan *query*

untuk intelijen bisnis atau kegiatan analitik lainnya. itu biasanya *update* dalam *batch*, tidak setiap kali transaksi yang terjadi adalah sistem sumber. Jadi *Data Warehouse* merupakan suatu sistem yang mengkonsolidasikan data secara periodik dari sistem-sistem yang ada OLPT (*On-Line Transaction Processing*) kedalam suatu penyimpanan dimensional. Pada umumnya, *Data Warehouse* menyimpan data *history* beberapa tahun dan akan dilakukan *query* untuk keperluan *business intelligence* atau aktifitas analisis lainnya. Perkembangan *data warehouse* digunakan sebagai sumber data untuk BI (*Business Intelligence*).

### Karakteristik *Data Warehousing*

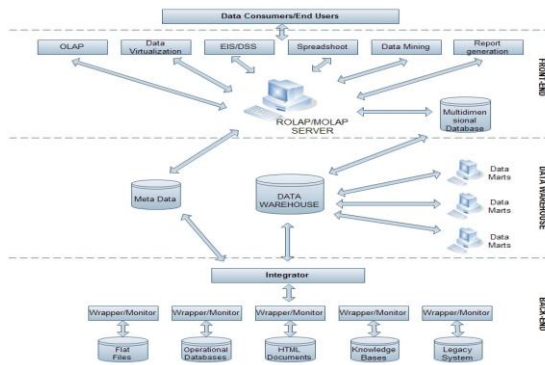
Berikut ini merupakan karakteristik utama dari *Data Warehouse* (Turban, 2005) :

1. Berorientasi subjek  
Data diorganisasi oleh subjek detail (misalkan berdasarkan pelanggan, jenis kebijakan, dan klaim dalam perusahaan) yang berisi hanya informasi yang relevan untuk mendukung keputusan. Orientasi subjek memungkinkan para pengguna untuk menentukan tidak hanya bagaimana bisnis mereka sedang berjalan, tetapi mengapa berjalan.
2. Terintegrasi  
Data pada sumber berbeda dapat diencode dengan cara yang berbeda. Data yang terintegrasi mengatasi inkonsistensi dan menyediakan istilah yang seragam di organisasi keseluruhan.
3. *Time variant*  
Data tidak disimpan untuk status saat ini. Mereka disimpan untuk lima atau sepuluh tahun atau lebih dan digunakan untuk tren, peramalan, dan perbandingan. Ada kakulasi sementara pada sebuah *Data Warehouse*. Waktu adalah dimensi penting yang harus didukung oleh semua *Data Warehouse*. Data untuk analisis dari berbagai sumber berisi berbagai poin waktu (missal harian, mingguan, bulanan).
4. *Nonvolatile*  
Sekali dimasukkan dalam *Warehouse*, data adalah *read-only*, mereka tidak bisa diubah atau diperbaharui. *Data Using* dibuang dan perubahan direkam sebagai data baru. Ini memungkinkan *Data Warehouse* untuk disesuaikan hampir secara eksklusif untuk akses data.
5. Ringkas  
Jika diperlukan, data oprasional dikumpulkan kedalam ringkasan– ringkasan.
6. Tidak ternormalisasi  
Data dalam sebuah *Data Warehouse* biasanya tidak dinormalisasi dan sangat redundan.
7. Sumber Semua data ada, baik eksternal dan internal.

8. Metadata

**Arsitektur Data Warehouse**

Umumnya sebuah sistem *data warehouse* terdiri dari tiga komponen utama, yaitu alat *back-end*, *data warehouse*, dan alat *front-end* seperti diilustrasikan pada Gambar 1.



**Gambar 1.** Arsitektur Data Warehouse (sumber : Sitompul,2008)

Alat *back-end* adalah sekumpulan perangkat lunak akuisisi data dengan tiga tugas utama yaitu meng-ekstrak data dari sumber-sumber eksternal, mengkonsolidasikan data ke dalam sebuah skema global, dan memuat data ke dalam *data warehouse*. Aktivitas-aktivitas ini umumnya dikenal sebagai proses ETL (*Extract, Transform, and Load*). Modul-modul yang bertanggung jawab untuk proses ekstraksi, konsolidasi dan pemuatan tersebut adalah *wrapper/monitor* dan *integrator*.

Modul *wrapper* bertugas mengubah sumber-sumber informasi dari format asalnya menjadi format dan model data yang digunakan oleh sistem *data warehouse*. Ada beberapa proses yang dilakukan pada saat berlangsungnya konversi, yaitu pemformatan ulang, pencucian, pengintegrasian, dan peringkasan sumber-sumber informasi. Peranan modul *monitor* adalah untuk mendeteksi secara otomatis setiap perubahan dalam sumber informasi dan melaporkannya kepada modul *integrator*, yang kemudian melakukan sederetan operasi seperti penyaringan, peringkasan, penggabungan, dan penginstalasian informasi ke dalam *data warehouse* [1,2].

**Keuntungan Data Warehouse**

*Data Warehouse* merupakan pendekatan untuk menyimpan data dimana sumber-sumber data yang heterogen (yang biasanya tersebar pada beberapa database OLTP) diimigrasikan untuk menyimpan data yang homogen dan terpisah. Keuntungan yang didapatkan dengan menggunakan *data warehouse* adalah [2,3] :

1. Data diorganisir dengan baik untuk *query* analisis dan sebagai bahan untuk pemrosesan transaksi.
2. Perbedaan antara struktur data yang homogen pada beberapa sumber yang terpisah dapat diatasi.
3. Aturan untuk transformasi data diterapkan untuk memvalidasi dan mengkonsolidasi data apabila data dipindahkan dari database OLTP ke *data warehouse*.
4. Masalah keamanan dan kinerja bisa dipecahkan tanpa perlu mengubah sistem produksi.

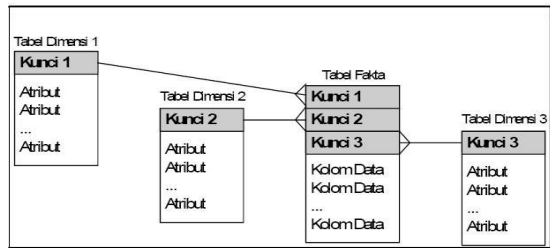
**Pengembangan Data Warehouse**

Proses imigrasi data ke sebuah *Data Warehouse* melibatkan ekstraksi data dari semua sumber relevan. Sumber data boleh terdiri dari file ekstraksi dari database OLTP, *spreadsheet*, database personal (misalnya *Microsoft Access*), atau file eksternal. *Data Warehouse* berisi banyak aturan bisnis yang menentukan hal-hal seperti bagaimana data akan digunakan, aturan peringkasan, standarisasi terhadap atribut terencode, dan aturan kalkulasi. Salah satu keuntungan *Data Warehouse* yang didesain dengan baik adalah aturan ini dapat disimpan pada sebuah *repository* metadata dan berlaku untuk *Data Warehouse* terpusat. Proses pemuatan ke dalam *Data Warehouse* dapat dilakukan melalui peranti transaksi data yang menyediakan sebuah GUI untuk membantu pengembangan dan pemeliharaan pengembangan aturan bisnis [2].

**Skema Star**

Desain *Data Warehouse* didasarkan atas konsep pemodelan dimensional. Pemodelan dimensional adalah suatu model berbasis pemanggilan yang mendukung akses *query*. Skema *star* adalah alat dimana pemodelan dimensional diterapkan. Skema *star* berisi sebuah table fakta pusat. Table fakta berisi atribut yang diperlukan untuk melakukan analisis putusan, atribut deskriptif yang digunakan untuk pelaporan *query*, dan *foreign key* untuk menghubungkan ketabel dimensi. Atribut analisis kebutuhan terdiri dari ukuran performa, metrik operasional, ukuran agregat, dan semua metrik lain yang diperlukan untuk menganalisis performa organisasi. Dengan kata lain, tabel fakta terutama menunjuk apa yang didukung oleh *data warehouse* untuk analisis keputusan. Sekeliling tabel fakta pusat dan hubungan via *foreign key* adalah tabel dimensi. Tabel dimensi berisi atribut yang menguraikan data yang dimasukkan dalam tabel fakta. Tabel dimensi menunjukkan bagaimana data akan dianalisis. Terdapat beberapa jenis skema, diantaranya skema star sederhana dalam skema ini, setiap tabel harus memiliki *primary key* yang terdiri dari satu kolom atau lebih. *Primary key*

dari tabel fakta terdiri dari satu atau lebih *foreign key*. *Foreign key* merupakan *primary key* pada tabel lain, seperti pada gambar 2.



Gambar 2. Skema Star Sederhana

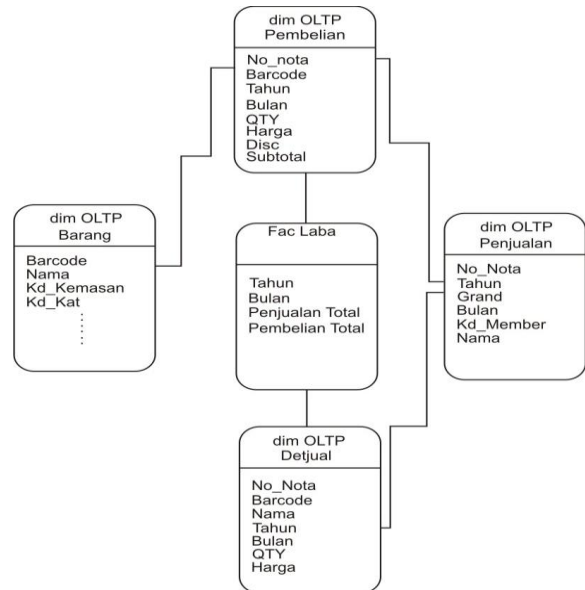
**HASIL PENELITIAN**

**Deskripsi Sistem**

Dalam penelitian ini dilakukan tahap pengolahan data untuk membuat data *warehouse*. Tahap yang pertama melakukan proses ETL (*extraction, transformasi, dan load*) dari sumber data yang ada dan data OLTP penjualan dijadikan *warehouse*. Proses ETL dilakukan di *Integration Services*. Proses *extraction* dilakukan dengan melakukan pemilihan data dari sumber data yang ada untuk dijadikan data *warehouse*. Sebelum data *warehouse* di simpan terlebih dahulu dilakukan proses *cleaning*, proses ini bertujuan untuk membersihkan data yang rangkap. Dengan adanya proses *cleaning* ini memungkinkan *user* untuk dapat meng-*update* data *warehouse* kapan saja, karena setiap kali *user* melakukan *update* maka data yang lama akan terhapus dan terganti dengan data yang baru tanpa menghilangkan data lama.

Proses *transformasi*, proses ini dilakukan untuk memindahkan data dan pemuatan (*loading*) dari sumber data ke data *warehouse* dengan menggunakan *script* tertentu. Tahapan selanjutnya dari data *warehouse* dapat dibuat laporan (*Report*) berupa *table* dan *graf chart* agar dapat melihat berapa besar mengalami laba rugi. Perangkat yang digunakan dalam pembuatan laporan adalah *Reporting Services* yang di integrasikan dengan *database SQL Server* yang digunakan data *warehouse*.

Skema yang digunakan dalam pemodelan data adalah *star scema*, dimana terdapat satu tabel fakta dan beberapa tabel dimensi. Table fakta berisi atribut – atribut yang diperlukan untuk membuat analisis, sedangkan tabel dimensi berisi tabel atribut yang menguraikan data yang ada dalam tabel fakta seperti pada gambar 3.

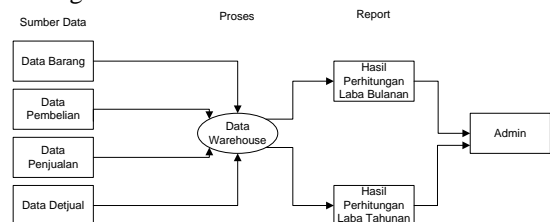


Gambar 3. Star Schema perhitungan laba

**Arsitektur Sistem**

Arsitektur menyediakan kerangka dengan mengidentifikasi dan memahami bagaimana data diproses menjadi *data warehouse* dan keluaran *report* yang nantinya dapat membantu dalam pengambilan keputusan.

Pada gambar 4.2 menjelaskan bahwa sumber data yaitu berupa data barang berupa *Barcode*, nama, kode kemasan, kode kat, kode subkat, kode dept, kode divisi, kode SP, isi, harga beli, harga jual, harga net, harga member, margin, *bestbuy*, *disc1*, *disc2*, *maxdisc*, *stat*, *discat*. Data pembelian berupa no nota, *barcode*, QTY, harga, *disc*, *subtotal*, tahun, bulan. Data penjualan berupa no nota, tahun, *grand*, kode member, nama, bulan. dan data detail penjualan berupa no nota, *barcode*, nama, QTY, harga, *subtotal*, tahun, bulan, kemudian diproses di *data warehouse* yang digunakan untuk *report* dan mengeluarkan hasil berupa perhitungan laba bulanan dan hasil berupa perhitungan laba tahunan.



Gambar 4. Arsitektur Sistem

**Perancangan Sistem**

Perancangan sistem merupakan bagian dari analisis sistem. Perancangan sistem adalah menentukan langkah-langkah dalam proses pengolahan data. Dalam penelitian ini dilakukan beberapa perancangan sistem, yaitu :

### Sumber Data

Sumber data berasal dari data penjualan yang ada di era yaitu berupa data barang, data penjualan, data pembelian, dan data detail penjualan mulai tahun 2006 hingga 2010

OLTP (*OnLine Transaction Processing*) yang digunakan adalah SQL Server 2005 yang berjalan di Windows Server 2003. Tabel yang digunakan hanya tabel yang berhubungan dengan penjualan untuk pembuatan laporan (*report*) saja. Tabel yang digunakan antara lain tabel barang yang berisi *barcode*, nama, kode kemasan, kode kat, kode subkat, kode dept, kode devisi, kode sp, isi, harga beli, harga jual, harga net, harga member, margin, *bestbuy*, *disc1*, *disc2*, *max-disc*, stat, dan disckat. Tabel pembelian berisi no nota, *barcode*, QTY, harga, *disc*, sub total, tahun, dan bulan. Tabel penjualan berisi no nota, tahun, grand, kode member, nama, dan bulan. Sedangkan tabel detail jual berisi no nota, *barcode*, nama, QTY, harga, sub total, tahun, dan bulan. *Database* OLTP yang sudah siap untuk ditransformasikan kedalam data *warehouse* kemudian untuk dilakukan proses ETL.

### Integration Service

*Integration service* merupakan mengintegrasikan data dari sumber data, dan menyalurkan data keluaran ke data *warehouse*. Proses ETL (*Extraction, Transforming, Loading*) merupakan proses yang akan dilalui dalam pembentukan data *warehouse*. Setelah sumber data dimasukkan dalam data *source* lalu di *extraction* untuk memilih data dari sumber data untuk membuat laporan hasil perhitungan laba yang akan di *transforming* ke dalam data *warehouse*. Pemilihan data atau *extraction* untuk pembuatan laporan perhitungan laba-rugi dapat dilakukan dengan *SQL Command*. *SQL Command* merupakan perintah SQL yang terdapat di dalam OLE DB Source.

Sebelum data ditransformasikan ke dalam data *warehouse* akan dilakukan proses *cleansing*. Proses *cleansing* ini adalah pembersihan data yang ada di data *warehouse* agar tidak terjadi penumpukan data yang kembar atau rangkap ketika diupdate oleh *user*. Proses *cleansing* ini terdapat di *Control Flow* yaitu *Start Execute SQL Task*. *Start Execute SQL task* dihubungkan dengan *Data FLOWTask*.

Proses berikutnya setelah *cleansing* adalah proses transformasi, dimana proses yang dilakukan adalah pengambilan data dari data *source* yang kemudian dimasukkan ke dalam data *warehouse*. Pada *Integration Service* proses transformasi yaitu dengan menjalankan *Execute Package* atau mengklik *start debugging*.

Setelah *package* dijalankan pada *Control Flow* area yaitu *Start Execute SQL Task* dan *Data FLOWTask* maka warna pada kotak menandakan

statusnya. Warna hijau menandakan berarti eksekusi berhasil. Jika bewarna kuning (*warning*) berarti eksekusi berjalan tetapi ada beberapa yang *error*, sedangkan merah berarti eksekusi *error* semua konfigurasi salah.

### Reporting Service

*Reporting Service* merupakan layanan untuk membuat *report*. Untuk melakukannya di buatuhkan *webbrowser* yang sudah tersedia di sistem operasi windows yaitu *Internet Explorer* yaitu sebagai media untuk menampilkan *report*. *Reporting Service* digunakan untuk membangun fasilitas *report* agar dapat digunakan untuk menganalisa data dan menampilkannya dalam bentuk tabel maupun *graf chart*.

Untuk membuat laporan atau *report* dibutuhkan data *sources*, data *sourceviews* dan *report models*. Data *sources* merupakan data sumber yang diambil dari data *warehouse* yang telah dibuat pada *Integration Service*. Data *sourch view* adalah sumber data yang dibutuhkan untuk menggambarkan suatu pandangan tentang data *warehouse* yang telah dibuat. Sedangkan *reportmodels* digunakan untuk membangun perancangan model kemudian menerbitkannya dalam bentuk laporan atau *report*. Sebuah model *report* bukanlah *report*. *Report Builder* digunakan untuk merancang laporan atau *report* sesuai kebutuhan *user*.

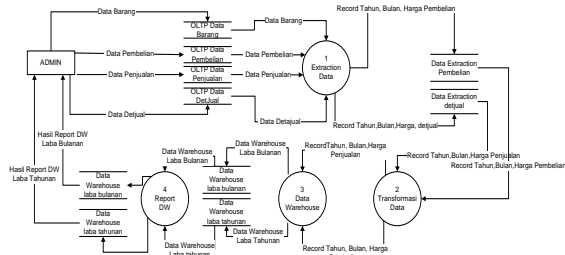
### Implementasi Sistem

Pada menu utama program SPK Pembelian *Notebook*, terdapat tombol akses untuk masuk ke menu pilihan. Sebagai calon pembeli, pengguna dapat melihat daftar *notebook*, konfigurasi batas nilai, dan menjalankan sistem rekomendasi. Sedangkan menu manajemen *notebook* hanya dapat diakses oleh administrator dengan melakukan proses login sebelumnya.

### DFD Level 1

Pada gambar 5. dijelaskan bahwa *user* memasukkan data barang (OLTP) berupa *Barcode*, nama, kode kemasan, kode kat, kode subkat, kode dept, kode devisi, kode SP, isi, harga beli, harga jual, harga net, harga member, margin, *bestbuy*, *disc1*, *disc2*, *maxdisc*, stat, disckat. Data pembelian(OLTP) berupa no nota, *barcode*, QTY, harga, *disc*, subtotal, tahun, bulan. Data penjualan(OLTP) berupa no nota, tahun, grand, kode member, nama, bulan. dan data detail penjualan (OLTP) berupa no nota, *barcode*, nama, QTY, harga, subtotal, tahun, bulan. Hasil data OLTP tersebut disimpan didalam *database* OLTP. Data OLTP selanjutnya akan di *extraction* atau dilakukan proses pemilihan data. Data yang di *extraction* adalah pembelian dan data detail penjualan, kemudian ditransformasi ke data

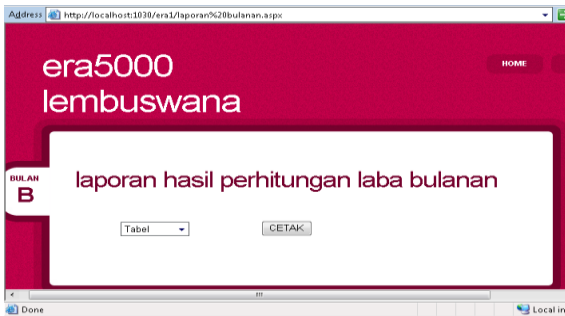
warehouse. Proses transformasi adalah proses pemindahan dan pemuatan (*loading*) data OLTP ke data warehouse. Dari data warehouse akan dibuat laporan atau *report* data hasil perhitungan laba bulanan dan hasil perhitungan laba tahunan.



Gambar 5. DFD level 1

**Pengujian Sistem Pada Laporan Bulanan**

Untuk melihat hasil laporan hasil perhitungan laba bulanan yang ada di era5000 lembuswana. *user* dapat memilih model laporan berupa tabel atau *graf chart* di halaman aplikasi laporan bulanan, lalu mengeksekusi tombol CETAK untuk proses. Halaman laporan data mahasiswa masuk dapat dilihat pada gambar 6.



Gambar 6. Halaman Laporan Bulanan

Pada saat *User* memilih laporan berupa tabel maka akan tampil rincian laba bulanan dalam bentuk tabel mulai dari tahun 2006 hingga tahun 2010 seperti yang terlihat pada gambar 6.

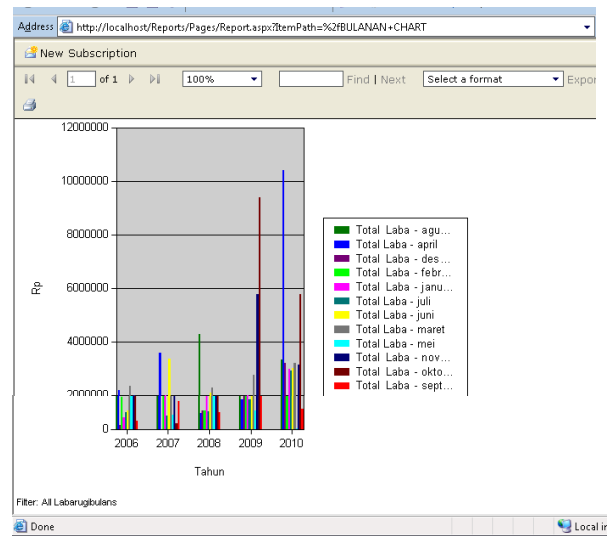
Tahun	Bulan	Total Pembelian	Total Penjualan	Total Laba
2006	agustus	1284012	3155000	1870988
	april	2236050	4401000	2164950
	desember	3001600	3136000	1344000
	februari	2332524	3530000	1197476
2007	januari	2603600	3023700	420100
	...	...	...	...

Gambar 6. Halaman Laporan Data Warehouse Bulanan Bentuk Tabel

Perintah SQL untuk proses perhitungan laba bulanan dan menyalurkan ketujuan yaitu data warehouse adalah:

```
SELECT Pembelian.tahun, Pembelian.bulan,
TotalPenjualan, TotalPembelian,
(TotalPenjualan-TotalPembelian) AS Jumlah
FROM (SELECT DISTINCT tahun, bulan,
SUM(harga) AS TotalPembelian FROM
dbo.pembelian GROUP BY tahun, bulan) AS
pembelian, (SELECT DISTINCT tahun, bulan,
SUM(Harga) AS TotalPenjualan FROM
dbo.detjual GROUP BY Tahun, bulan) AS
penjualan WHERE pembelian.tahun =
penjualan.tahun AND pembelian.bulan =
penjualan.bulan
```

*user* memilih laporan berupa *graf chart* maka akan tampil laba bulanan dalam bentuk *graf chart* yang mulai dari tahun 2006 sampai dengan tahun 2010 seperti pada gambar 7.



Gambar 7. Halaman Laporan Data Warehouse Bulanan Graf Chart

**KESIMPULAN**

Dari hasil penelitian mengenai laporan hasil perhitungan laba rugi era5000 lembuswana dengan menggunakan *warehouse*, dapat disimpulkan antara lain :

1. Penggunaan sumber data sebagai OLTP dan dilakukan proses ETL (*extraction, transformation, dan load*) ke data *warehouse*. Kemudian data *warehouse* berisi ringkasan dan data keseluruhan dari sumber data yang ada.
2. Data yang telah diujikan sebanyak 1600 (seribu enam ratus) data dan merupakan contoh data yang digunakan pada Era 5000 Lembuswana.
3. *Report* data *warehouse* hanya dapat ditampilkan melalui *browser Internet Explorer*.
4. Berdasarkan tujuannya Data yang dihasilkan adalah data *read only*, yang tidak dapat dilakukan proses *delete* dan *edit*.

**DAFTAR PUSTAKA**

- [1] Sitompul, S.O. 2006. Data Warehouse Inisiatif di Universitas Sumatra Utara. Skripsi: Teknologi Informasi, Universitas Sumatra Utara.
- [2] Turban, E., Arosan, J., dan Liang, P.T., 2005. Decision Support System (Sistem pendukung keputusan dan sistem cerdas). Yogyakarta. Andi.
- [3] Nugroho, A.D., Johan, T., Hoetama, J.T., 2008. Aplikasi Data Warehouse Untuk Analisis Penjualan Mobil Berbasis Multidimensional Modeling (MDM) dan Star Schema Desain (studi kasus: PT. Asco Automotive) Jurnal Teknologi Informasi. Vol.5 No. 2. Pp 186-197.