PEMBANGUNAN PERANGKAT LUNAK DATA MART PADA PT. MATAHARI SENTOSA

Ihsan Mukhlish Shidiq[1]

[1] Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia
Jl. Dipatiukur No 112-116 Bandung - Indonesia
E-mail : ihsanmukhlishshidiq@gmail.com

ABSTRAK

PT. Matahari Sentosa merupakan perusahaan resleting. Perusahaan ini dipercaya sebagai *supplier* bagi produsen lain. Saat ini dalam pembuatan laporan akhir produksi yang multidimensi menjadi kurang efektif belum adanya suatu sistem untuk mendukung dalam pembuatan laporan yang multidimensi. Dalam menyajikan informasi yang memiliki periode waktu akan membutuhkan waktu yang cukup lama untuk menyajikannya.

Berdasarkan permasalahan yang ada pada divisi produksi perlu dibangun sebuah perangkat lunak *data mart* yang dapat memudahkan dalam mendapatkan informasi strategis dengan cepat sehingga dapat digunakan untuk perencanaan jangka panjang, dapat menyajikan informasi yang multidimensi dan ringkas, mempercepat proses penganalisaan sehingga dapat memaksimalkan suatu keputusan yang dibuat oleh manajer produksi untuk penanganan stok produk dan penanganan bahan baku untuk produksi selanjutnya.

Dengan dibangunnya perangkat lunak *data mart* ini dapat diambil kesimpulan yaitu, memberikan kemudahan dalam mendapatkan informasi strategis dengan cepat yang diperlukan oleh manajer produksi PT. Matahari Sentosa untuk perencanaan jangka panjang, dapat menyajikan informasi yang multidimensi dan ringkas, serta mempercepat proses penganalisaan sehingga dapat memaksimalkan suatu keputusan yang dibuat oleh manajer produksi untuk penanganan stok produk yang sudah diproduksi dan penanganan bahan baku untuk produksi selanjutnya, dan membantu dalam pembuatan laporan akhir produksi yang multidimensi yang dibutuhkan oleh manajer produksi di PT. Matahari Sentosa.

Kata kunci : *Data Mart*, *Data Warehouse*, Skema *Constellation*

1. PENDAHULUAN

PT. Matahari Sentosa merupakan perusahaan resleting. Perusahaan ini dipercaya sebagai *supplier* bagi produsen lain, seperti produsen pembuat tas, jaket, celana, dan pembuatan barang yang membutuhkan resleting

Saat ini pembuatan laporan akhir yang multidimensi atau memiliki periode waktu menjadi kurang efektif dan efisien, karena belum ada suatu sistem untuk mendukung dalam pembuatan laporan yang multidimensi. Dalam menyajikan informasi yang memiliki periode waktu akan membutuhkan waktu yang cukup lama dalam menyajikannya dikarenakan oleh pencarian data dan pemrosesannya yang lambat serta belum adanya suatu sistem yang dapat menyajikan informasi yang cepat dan detail. Untuk itu perlu dibangunnya *data mart* untuk memudahkan pihak manajer produksi untuk mendapatkan informasi strategis dengan cepat sehingga dapat digunakan untuk perencanaan jangka panjang dan juga dapat menyajikan informasi yang multidimensi dan ringkas, mempercepat proses penganalisaan sehingga dapat memaksimalkan suatu keputusan yang dibuat oleh manajer produksi untuk penanganan stok produk dan penanganan bahan baku untuk produksi selanjutnya.

Tujuan dari membangun perangkat lunak *data mart* pada PT. Matahari Sentosa ini adalah untuk :

1. Memudahkan pihak manajer produksi untuk mendapatkan informasi strategis dengan cepat sehingga dapat digunakan untuk perencanaan jangka panjang serta dapat menyajikan informasi yang multidimensi dan ringkas, mempercepat proses penganalisaan sehingga dapat memaksimalkan suatu keputusan yang dibuat oleh manajer produksi untuk penanganan stok produk yang sudah diproduksi dan penanganan bahan baku untuk produksi selanjutnya.
2. Membantu pihak manajer produksi dalam pembuatan laporan akhir produksi yang multidimensi.

Batasan masalah yang diterapkan dalam pembangunan perangkat lunak *data mart* ini antara lain, yaitu :

1. Data yang digunakan berasal dari Divisi Produksi, yaitu data produksi, pencelupan, stok produksi, dan stok produk.
2. Data untuk pengujian *data mart* yaitu tahun 2010 sampai dengan tahun 2014.

3. Proses pembuatan *data mart* dengan proses ETL (*Extract Transform Loading*).
4. Pengujian *data mart* menggunakan OLAP (*On-Line Analytical Processing*).
5. DBMS menggunakan SQL Server 2012.
6. Pembangunan perangkat lunak menggunakan Visual Studio 2012.
7. Analisis dan perancangan perangkat lunak menggunakan analisis berorientasi objek.

1.1 Data Mart

Data mart merupakan suatu bagian dari *data warehouse* yang mendukung pembuatan laporan dan analisis pada suatu unit. *Data mart* berisi informasi yang relefan bagi pengguna yang ingin mengambil keputusan.

Ada empat tugas yang dapat dilakukan dengan adanya *data mart* [1], empat tugas tersebut adalah sebagai berikut :

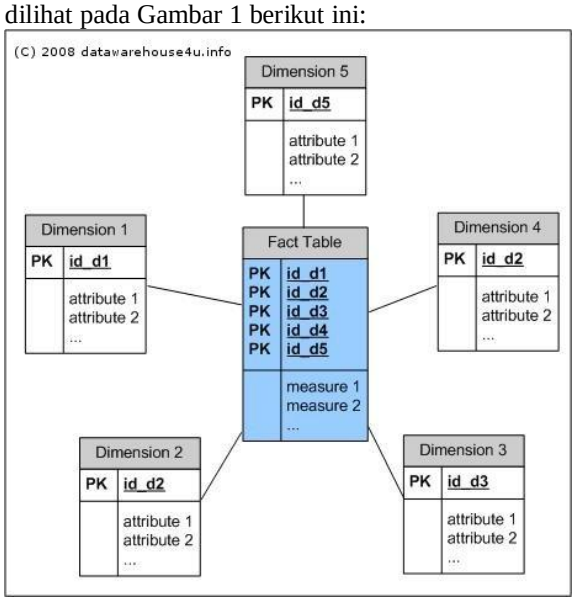
1. Pembuatan laporan
   Pembuatan laporan merupakan salah satu kegunaan *data mart* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan perhari, perbulan, pertahun, dan jangka waktu kapanpun yang diigninkan.
2. *On-Line Analytical Processing* (OLAP)
   Dengan adanya *data mart*, semua informasi baik detail maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah didapat. OLAP merupakan konsep data multidimensi dan memungkinkan pada pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL. Fasilitas lain adalah *roll-up* dan *drill-down*. *Drill-down* adalah kemampuan untuk melihat detail dari suatu informasi dan *roll-up* adalah kebalikannya.
3. *Data Mining*
   Data mining merupakan proses untuk menggali (*mining*) pengetahuan dan informasi baru dari data yang berjumlah banyak pada *data mart*.
4. Proses informasi *executive*
   Data mart dapat membuat ringkasan informais yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan *data mart* segala laporan telah diringkas dan dapat pula mengetehaui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan.

1.2 Model Dimensional Data Mart

Model dimensional pada *data mart* terdiri dari tabel fakta dan tabel dimensi. Tabel fakta merupakan tabel yang berisi kumpulan *foreign key* dari *primary* key yang terdapat pada masing-masing tabel dimensi, sedangkan tabel dimensi merupakan tabel yang berisi data detail yang menjelaskan *foreign key* yang terdapat pada tabel fakta. Ada beberapa model skema yang terdapat pada pemodelan *data mart*, yaitu skema *star*, skema *snowflake*, dan skema *constellation*. Penjelasan dari masing-masing model skema adalah sebagai berikut :

1. Skema Bintang (*Star Schema*)

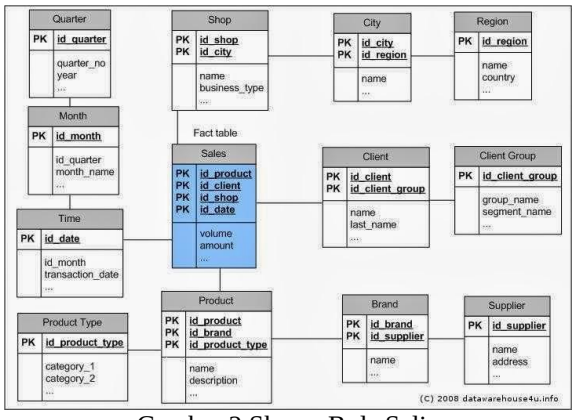
Skema ini mengikuti bentuk bintang, dimana terdapat satu tabel fakta di pusat bintang dengan beberapa tabel dimensi yang mengelilinginya. Semua tabel dimensi berhubungan dengan ke tabel fakta. Tabel fakta memiliki beberapa *primary key* dalam tabel dimensi. Contoh skema bintang dapat dilihat pada Gambar 1 berikut ini:



Gambar 1 Skema Bintang

2. Skema Bola Salju (*Snowflake Schema*)

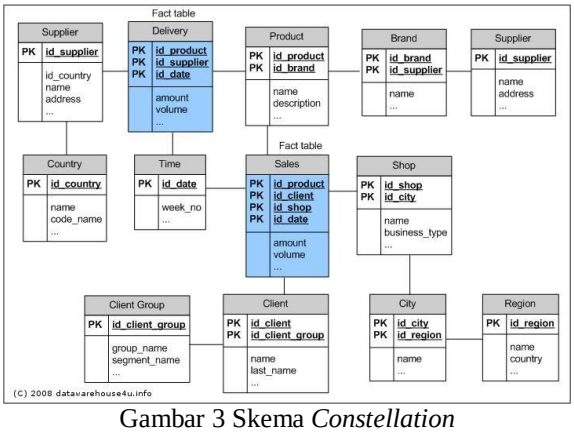
Skema bola salju merupakan perluasan dari skema bintang dengan tambahan beberapa tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta. Tabel dimensi tersebut berhubungan dengan tabel dimensi yang lain. Contoh skema bola salju dapat dilihat pada Gambar 2 berikut ini:



Gambar 2 Skema Bola Salju

3. Skema *Constellation*

Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi. Contoh skema c*onstellation* dapat dilihat pada Gambar 3 berikut ini:



Gambar 3 Skema *Constellation*

1.3 ETL (*Extract, Transform, Loading*)

Proses ETL (*Extract, Transform, Loading*) merupakan proses yang harus dilalui dalam pembentukan *data mart* [2]. ETL merupakan fase pemrosesan data dari sumber data kemudian masuk ke dalam *data mart*. Tujuan ETL adalah mengumpulkan, menyaring, mengolah dan menggabungkan data dari berbagai sumber untuk disimpan ke dalam *data mart*. Berikut adalah penjelasan dari tiap proses ETL :

1. Ekstraksi Data (*Extract*)

Ekstraksi data adalah proses pengambilan data dari berbagai sistem operasional, baik menggunakan *query* atau aplikasi ETL. Terdapat beberapa fungsi ekstraksi data, yaitu :

a. Ekstraksi data secara otomatis dari aplikasi sumber.
b. Penyaringan atau seleksi data hasil ekstraksi.
c. Pengiriman data dari berbagai *platform* aplikasi ke sumber data.
d. Perubahan format *layout* data dari format aslinya.
e. Penyimpanan dalam *file* sementara untuk penggabungan dengan hasil ekstraksi dari sumber lain.

2. Transformasi Data (*Transformation*)

Transformasi adalah proses dimana data hasil ekstraksi disaring dan diubah sesuai dengan aturan bisnis yang berlaku. Langkah-langkah dalam transformasi data adalah sebagai berikut :

a. Memetakan data input dari skema data aslinya ke skema *data mart*.
b. Melakukan konversi tipe data.
c. Membersihkan dan membuang data yang sama (duplikasi).
d. Memeriksa referensi data.
e. Mengisi nilai-nilai yang kosong dengan nilai *default*.
f. Menggabungkan data.

3. Pengisian Data (*Loading*)

Pengisian data adalah proses memasukkan data yang didapatkan dari hasil transformasi ke dalam *data mart*. Cara untuk memasukkan data adalah dengan menjalankan *SQL script* secara periodik.

1.4 OLAP (*On-Line Analytical Processing*)

OLAP (*On-Line Analytical Processing*) adalah teknologi yang memproses data ke dalam struktur multidimensi, menyediakan jawaban yang cepat untuk *query* analisis yang kompleks dengan tujuan untuk mengorganisir sejumlah data yang besar, agar bisa dianalisa dan dievaluasi dengan cepat serta menyediakan kecepatan dan fleksibelitas untuk mendukung analisa dalam waktu nyata [3].

Ada beberapa karakteristik OLAP, yaitu :

1. Mengijinkan pelaku bisnis melihat data dari sudut pandang *logical* dan multidimensional pada *data warehouse*.
2. Memfasilitasi *query* yang kompleks dan analisa bagi pengguna.
3. Mengijinkan pengguna melakukan *drill-down* untuk menampilkan data yang lebih detail atau *roll-up* untuk agregasi dari suatu dimensi atau beberapa dimensi.
4. Menyediakan proses kalkulasi dan perbandingan data.
5. Menampilkan hasil dalam bentuk tabel atau grafik.

Keuntungan dari OLAP, yaitu :

1. Meningkatkan produktivitas pemakai akhir bisnis, pengembang IT, dan keseluruhan organisasi.
2. Pengawasan yang lebih dan akses tepat waktu terhadap informasi strategis dapat membuat pengambilan keputusan lebih cepat.
3. Mengurangi pengembangan aplikasi bagi staf IT dengan membuat pemakaian akhir dapat mengubah skema dan membuat model sendiri.
4. Penyimpanan pengawasan organisasi melalui integritas data koorporasi sebagai aplikasi OLAP tergantung pada *data warehouse* dan sistem OLTP untuk memperbaharui sumber tingkatan data.

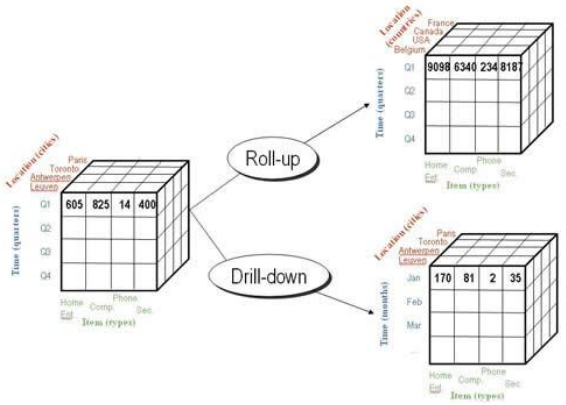
OLAP dapat digunakan untuk melakukan seperti [3]:

1. Konsolidasi (*roll-up*)

Konsolidasi melibatkan pengelompokan data.

2. *Drill-down*

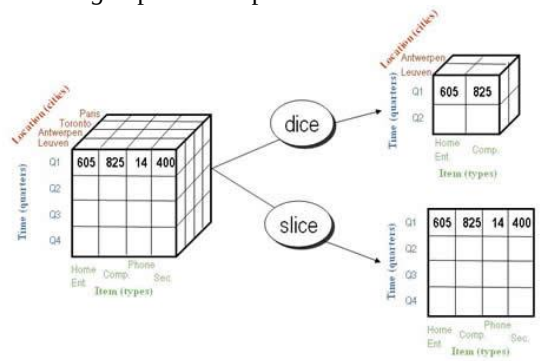
Suatu bentuk yang merupakan kebalikan dari konsolidasi untuk menjabarkan data yang ringkas menjadi data yang lebih detail. Gambaran untuk *roll-up* dan *drill-down* dapat dilihat pada Gambar 4 berikut ini:



Gambar 4 *Roll-up* dan *Drill-down*

3. *Slicing* dan *dicing*

Menjabarkan pada kemampuan untuk melihat data dari sudut pandang. Gambaran untuk *slicing* dan *dicing* dapat dilihat pada Gambar 5 berikut ini:



Gambar 5 *Slicing* dan *Dicing*

1.5 SSIS (*SQL Server Integration Service*)

SSIS (*SQL Server Integration Services*) adalah suatu platform untuk membangun sistem yang handal untuk integrasi data, *extraction*, *transformation*, dan *loading* yang digunakan pada *data warehousing* [4]. SSIS menawarkan solusi dalam menghadapi permasalahan data integrasi. Selain itu, *tools* ini membatu untuk menigkatkan efisiensi waktu pembuatan.

Arsitektur *SQL Server Integration Services* secara umum berisi berbagai macam komponen, diantaranya yaitu :

1. *SSIS Deginer*. Merupakan *tools* yang digunakan untuk membuat dan mengatur paket *integration service*. Pada *SQL Server 2012*, *tools* ini sudah terintegrasi dengan *Visual Studio 2010* yang merupakan bagian *project Bussiness Intelegence*.
2. *Runtime Engine*. Komponen ini berguna untuk menjalankan semua paket SSIS yang sudah dibuat.
3. *Task* dan *executable binary*.
4. *Data Flow Engine* dan *Data Flow*. Komponen *data flow* merupakan enkapsulasi *data flow engine* yang menyediakan *buffer* didalam memori dan bertugas memindahkan data dari sumber data ke tujuan data. Sedangkan *data flow* merupakan sumber data, tujuan data, dan transformasi.
5. *Integration Services service*. Memungkinkan *SQL Server Management Studio* dapat digunakan untuk melakukan monitoring paket SSIS dan mengatur SSIS *storage* yang digunakan.
6. *SQL Server Import* dan *Export Wizard*. *Tools* ini digunakan untuk melakukan *copy* data dari sumber ke tujuan data.

2 ISI PENELITIAN

Terdapat beberapa tahapan analisis dalam pembangunan perangkat lunak *data mart* ini, yaitu:

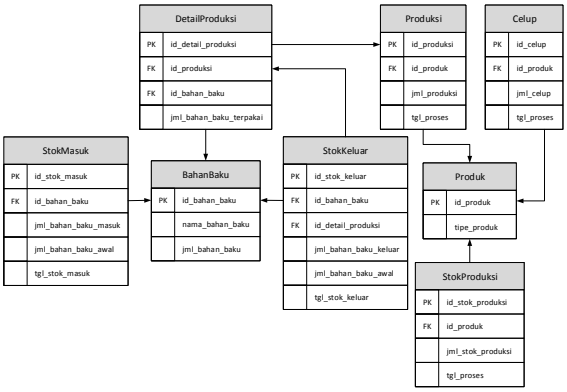
2.1 Analisis Kebutuhan Informasi

Analisis kebutuhan informasi merupakan tahap untuk menganalisis apa saja yang dibutuhkan oleh PT. Matahari Sentosa untuk *data mart* yang akan dibangun. Informasi yang akan disajikan secara detail. Berdasarkan hasil wawancara dengan manajer produksi, informasi yang dibutuhkan untuk pembangunan *data mart* adalah sebagai berikut :

1. Informasi jumlah produk yang diproduksi pertahun, perbulan, dan perhari.
2. Informasi jumlah produk yang diproduksi berdasarkan tipe produk pertahun, perbulan dan perhari.
3. Informasi jumlah produk yang sudah diproses pencelupan pertahun, perbulan, dan perhari.
4. Informasi jumlah produk yang sudah diproses pencelupan berdasarkan tipe produk pertahun, perbulan dan perhari.
5. Informasi jumlah pemakaian bahan baku pertahun, perbulan, dan perhari.
6. Informasi jumlah pemakaian bahan baku berdasarkan bahan baku tertentu pertahun, perbulan dan perhari.

2.2 Sumber Data

Database OLTP yang sekarang berada di PT. Matahari Sentosa menjadi sumber data untuk membangun *data mart*. Diagram relasi OLTP PT. Matahari Sentosa dapat dilihat pada Gambar 6 berikut ini:



Gambar 6 Diagram Relasi OLTP PT. Matahari Sentosa

2.3 Analisis Arsitektur Data Mart

Arsitektur yang digunakan untuk pembangunan *data mart* adalah menggunakan model *two-layer architecture*. Analisis arsitektur ini terbagi dalam empat *layer*, yaitu analisis *source layer*, analisis *data stagging*, analisis *data mart layer*, dan analisis menggunakan OLAP.

1. Analisis *Source Layer*

Pada lapisan ini, data masih berupa data operasional. Sumber data yang digunakan dalam pembangunan *data mart* ini sudah berupa data *logic* yang ada di *database*.

2. Analisis *Data Stagging*

Pada lapisan ini, data operasional akan diekstrak dengan proses ETL (*Extract Transform Loading*) ke dalam *data mart*.

a. *Extract*

Proses ini merupakan pemilihan data dari sumber data untuk pembuatan *data mart*, yaitu tabel produk, tabel produksi, tabel celup, tabel bahan baku, tabel stok keluar, dan tabel stok produksi serta tabel yang tidak dijadikan untuk pembuatan *data mart*, yaitu tabel detail produksi dan tabel stok masuk karena tidak dibutuhkan dalam kebutuhan informasi. Atribut-atribut yang ada pada tabel yang akan diekstrak tidak ada perubahan menambah atau mengurangi atribut-atributnya, masih tetap sama dengan sumber data. Proses ekstraksi data dari sumber data ke dalam *data mart* adalah sebagai berikut :

Tabel 1 *Extract*

No	Nama Tabel	Field
1	Tabel Produk	id_produk
		tipe_produk
2	Tabel Produksi	id_produksi
		id_produk
		jml_produksi
		tgl_proses
3	Tabel Celup	id_celup
		id_produk
		jml_celup
		tgl_proses
4	Tabel Bahan Baku	id_bahan_baku
		nama_bahan_baku
		jml_bahan_baku
5	Tabel Stok Keluar	id_stok_keluar
		id_bahan_baku
		id_detail_produksi
		jml_bahan_baku_keluar
		jml_bahan_baku_awal
		tgl_stok_keluar
6	Tabel Stok Produksi	id_stok_produksi
		id_produk
		jml_stok_produksi
		tgl_proses

b. *Transform*

Proses *transform* yang dilakukan terdiri dari 2 proses, yaitu :

1) *Cleaning*

Proses *cleaning* dilakukan untuk membersihkan data yang tidak digunakan dari tabel yang sudah diekstrak, yaitu menghilangkan *field* yang tidak terpakai. Berikut merupakan nama *field* yang dihilangkan dalam proses *cleaning* :

a) Pada tabel produksi tidak memerlukan *field* id_produksi dan tgl_proses.
b) Pada tabel celup tidak memerlukan *field* id_celup dan tgl_proses.
c) Pada tabel bahan baku tidak memerlukan *field* jml_bahan_baku.
d) Pada tabel stok keluar tidak memerlukan *field* id_stok_keluar, id_detail_produksi, dan tgl_stok_keluar.
e) Pada tabel stok produksi tidak memerlukan *field* id_stok_produksi dan tgl_proses.

2) *Conditioning*

Proses *conditioning* dilakukan untuk mengubah format dari data oprasional menjadi format *data mart*. Tabel di-*condotioning* dilakukan pada tabel produksi, tabel celup, tabel stok keluar, dan tabel stok produksi. Proses *conditioning* dilakukan untuk memastikan tidak adanya redudansi data, sehingga pada suatu tabel fakta bisa memiliki lebih dari satu tabel dimensi. Pada Tabel 2 akan dijelaskan contoh dari tahapan *conditioning* pada tabel produksi.

Tabel 2 *Conditioning* Tabel Produksi

Tabel Produksi	
No	**Field**
1	id_produk
2	jml_produksi
3	tgl_proses

→

Fact_produksi	
No	**Field**
1	id_produk
2	jml_produksi
3	id_waktu

Tabel Produksi	
tgl_proses	date

→

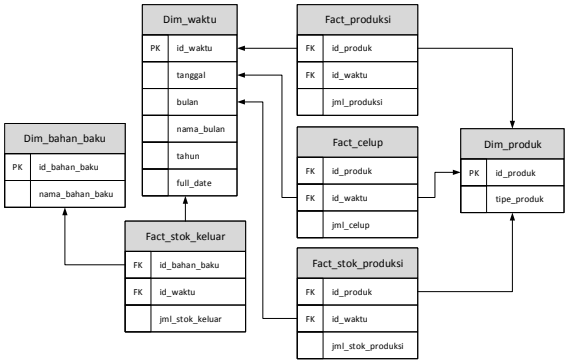
Dim_waktu	
id_waktu	integer
tanggal	integer
bulan	integer
nama_bulan	nvarchar(50)
tahun	integer
full_date	date

c. *Loading*

Setelah data di *extract* dan di *transform*, selanjutnya data tersebut dimasukkan ke dalam *data mart*. Proses *loading* pada aplikasi *data mart* akan dilakukan secara otomatis setelah proses *transform* selesai. Teknik yang digunakan adalah *update*, dimana proses ini akan langsung meng-*update data mart* tanpa merubah data yang sudah ada.

3. Analisis *Data Mart Layer*

Pada lapisan ini, data yang sudah melalui proses ETL akan disimpan pada sebuah penyimpanan *logic* yang tersentralisasi, yaitu *data mart*. Skema yang digunakan dalam pembangunan *data mart* adalah skema *constellation*. Skema *constellation* dalam pembangunan *data mart* dapat dilihat pada Gambar 7 berikut ini:



Gambar 7 Skema *Constellation*

Tabel 3 menunjukkan jenis dari masing-masing tabel yang ada pada skema relasi yang ditunjukkan pada Gambar 7.

Tabel 3 Penjelasan Skema *Data Mart*

No	Nama Tabel	Jenis Tabel
1	Fact_Produksi	Fakta
2	Fact_Celup	Fakta
3	Fact_Stok_Keluar	Fakta
4	Fact_Stok_Produksi	Fakta
5	Dim_Produk	Dimensi
6	Dim_Bahan_Baku	Dimensi
7	Dim_Waktu	Dimensi

2.4 Spesifikasi Kebutuhan Perangkat Lunak

Analisis spesifikasi kebutuhan perangkat lunak berisi deskripsi dari kebutuhan perangkat lunak yang akan dibangun baik kebutuhan fungsional maupun kebutuhan non fungsional. Tabel spesifikasi kebutuhan perangkat lunak dapat dilihat pada Tabel 4 dan Tabel 5.

Tabel 4 Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode Kebutuhan	Kebutuhan
SKPL-F-1	Administrator dan manajer produksi dapat mengakses aplikasi.
SKPL-F-2	Sistem dapat melakukan *update* data pada *data mart*.
SKPL-F-3	Sistem dapat melakukan proses ETL (*Extract Transform Loading*).
SKPL-F-4	Manajer produksi dapat melakukan analisis terhadap *data mart*.
SKPL-F-5	Sistem dapat menampilkan hasil analisis berupa tabel dan grafik.
SKPL-F-6	Sistem dapat memberikan laporan tercetak.

Tabel 5 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional

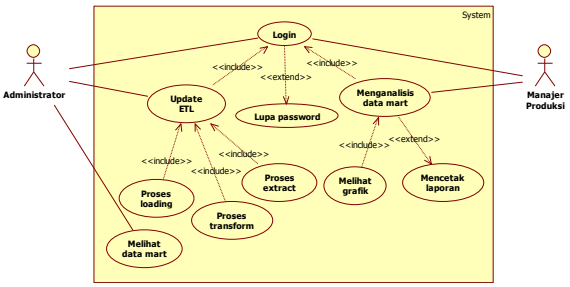
Kode Kebutuhan	Kebutuhan
SKPL-NF-1	Sistem dapat berjalan pada *platfrom windows*.
SKPL-NF-2	Sistem dapat berjalan pada *hardware* berspesifikasi menengah.
SKPL-NF-3	Bahasa pemrograman yang digunakan adalah C#.

2.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menguraikan proses kegiatan dan kebutuhan yang akan diterapkan pada sistem yang akan dibangun.

2.5.1 Use Case Diagram

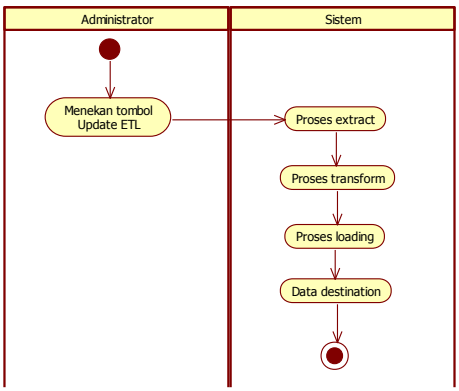
Use case diagram merupakan gambaran fungsionalitas atau perilaku dari sistem. *Use case diagram* pada perangkat lunak dapat dilihat pada Gambar 8 berikut ini:



Gambar 8 *Use Case Diagram*

2.5.2 Activity Diagram

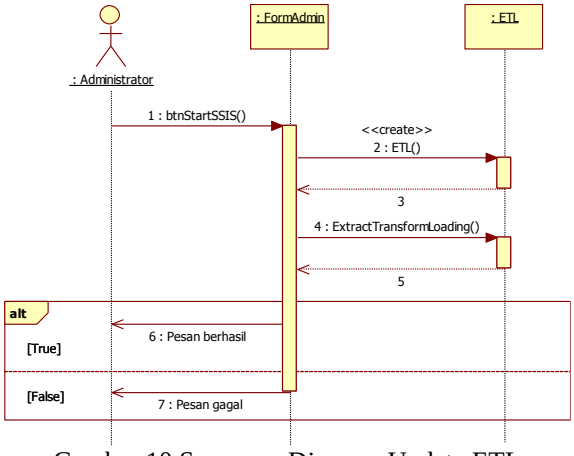
Activity diagram memodelkan aliran kerja dari urutan aktivitas dalam suatu proses yang mengacu pada *use case diagram* yang ada. *Activity diagram* dari *update* ETL dapat dilihat pada Gambar 9 berikut ini:



Gambar 9 *Activity Diagram Update ETL*

2.5.3 Sequence Diagram

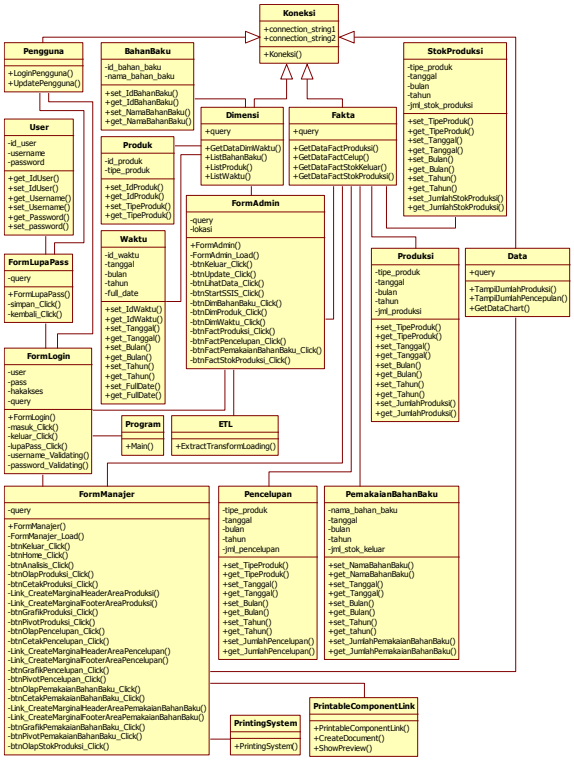
Sequence diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. *Sequence diagram* untuk *update* ETL dapat dilihat pada Gambar 10 berikut ini:



Gambar 10 Sequence Diagram Update ETL

2.5.4 Class Diagram

Class diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. *Class diagram* perangkat lunak yang akan dibangun dapat dilihat pada Gambar 11 berikut ini:



Gambar 11 Class Diagram

2.6 Implementasi Basis Data

Pembuatan basis data dilakukan dengan menggunakan aplikasi DBMS *Microsoft SQL Server 2012*. Implementasi basis data dalam bahasa *SQL* adalah sebagai berikut :

Tabel 6 Implementasi Tabel dim_produk

Dimensi Produk

```
CREATE TABLE [dbo].[dim_produk](
	[id_produk] [int] NOT NULL,
	[tipe_produk] [varchar](20) NOT NULL,
 CONSTRAINT [PK_dim_produk] PRIMARY KEY
CLUSTERED
(
	[id_produk] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE =
OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Tabel 7 Implementasi Tabel dim_waktu

Dimensi Waktu

```
CREATE TABLE [dbo].[dim_waktu](
	[id_waktu] [int] IDENTITY(1,1) NOT NULL,
	[tanggal] [int] NOT NULL,
	[bulan] [nvarchar](50) NOT NULL,
	[tahun] [int] NOT NULL,
	[full_date] [date] NOT NULL,
 CONSTRAINT [PK_dim_waktu] PRIMARY KEY CLUSTERED
(
	[id_waktu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE =
OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Tabel 8 Implementasi Tabel fact_produksi

Fakta Produksi

```
CREATE TABLE [dbo].[fact_produksi](
	[id_produk] [int] NOT NULL,
	[jml_produksi] [int] NOT NULL,
	[id_waktu] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[fact_produksi]  WITH CHECK
ADD  CONSTRAINT [FK_fact_produksi_dim_produk]
FOREIGN KEY([id_produk])
REFERENCES [dbo].[dim_produk] ([id_produk])
GO

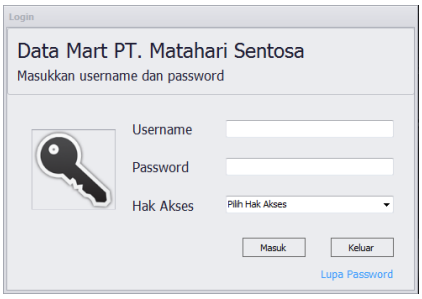
ALTER TABLE [dbo].[fact_produksi] CHECK
CONSTRAINT [FK_fact_produksi_dim_produk]
GO

ALTER TABLE [dbo].[fact_produksi]  WITH CHECK
ADD  CONSTRAINT [FK_fact_produksi_dim_waktu]
FOREIGN KEY([id_waktu])
REFERENCES [dbo].[dim_waktu] ([id_waktu])
GO

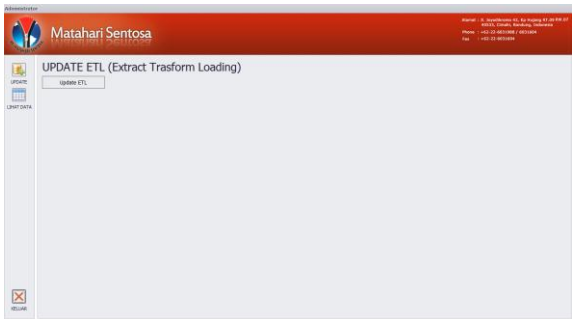
ALTER TABLE [dbo].[fact_produksi] CHECK
CONSTRAINT [FK_fact_produksi_dim_waktu]
GO
```

2.7 Implementasi Antar Muka

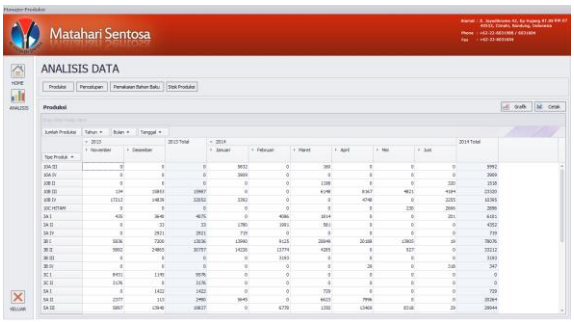
Implementasi antar muka pada perangkat lunak ini terdiri dari beberapa halaman. Berikut implementasi antar muka perangkat lunak *data mart*.



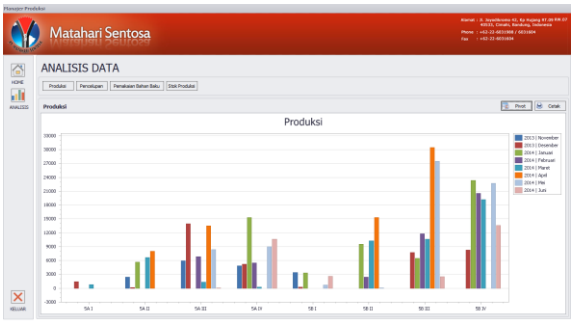
Gambar 12 Halaman Login



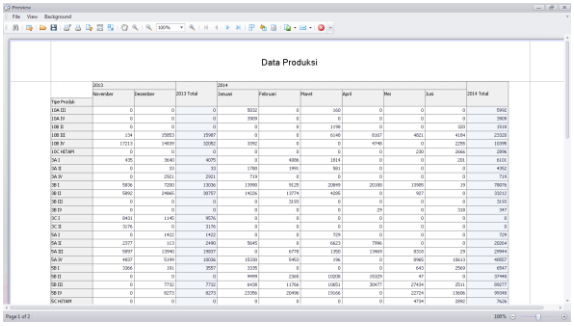
Gambar 13 Halaman *Update* ETL



Gambar 14 Halaman Menganalisis *Data Mart*



Gambar 15 Halaman Melihat Grafik



Gambar 16 Halaman Mencetak Laporan

3 PENUTUP

3.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai analisis, perancangan, implementasi, dan pengujian terhadap perangkat lunak *data mart* untuk divisi produksi di PT. Matahari Sentosa ini, maka dapat disimpulkan sebagai berikut :

1. Penerapan *Data Mart* memberikan kemudahan dalam mendapatkan informasi strategis dengan cepat yang diperlukan oleh manajer produksi PT. Matahari Sentosa untuk perencanaan jangka panjang, dapat menyajikan informasi yang multidimensi dan ringkas, serta mempercepat proses penganalisaan sehingga dapat memaksimalkan suatu keputusan yang dibuat oleh manajer produksi untuk penanganan stok produk yang sudah diproduksi dan penanganan bahan baku untuk peoruksi selanjutnya.
2. Penerapan *Data Mart* membantu dalam pembuatan laporan akhir produksi yang multidimensi yang dibutuhkan oleh manajer produksi di PT. Matahari Sentosa.

3.2 Saran

Berikut ini saran yang dapat digunakan dalam pengembangan perangkat lunak *data mart* lebih lanjut adalah sebagai berikut :

1. Sistem tidak hanya menganalisis Divisi Produksi saja, tetapi menganalisis seluruh divisi yang ada di PT. Matahari Sentosa guna untuk memenuhi informasi strategis perusahaan.
2. Untuk pengembangan perangkat lunak *data mart* yang sudah dibuat dapat dikembangkan kembali untuk pembuatan *data mart* pada divisi lainnya atau *data warehouse* untuk PT. Matahari Sentosa.

DAFTAR PUSTAKA

[1] M. Golfarelli and S. Rizzi, Data Warehouse Design, Modern Principles and Methodologies, New York: Mc Graw Hill, 2009.

[2] R. Kimball and M. Ross, The Data Warehouse Toolkit, Indianapolis: John Wiley & Sons, Inc, 2013.

[3] M. Y. Pusadan, Rancang Bangun Data Warehouse, Yogyakarta: Graha Ilmu, 2013.

[4] B. Knight, D. Knight, M. Davis and W. Snyder, Microsoft SQL Server 2012 Integration Services, Indianapolis: John Wiley & Sons, Inc, 2013.