

## PERANCANGAN DAN PEMBUATAN DATA WAREHOUSE PADA PERPUSTAKAAN STMIK AMIKOM YOGYAKARTA

Armadyah Amborowati  
STMIK AMIKOM Yogyakarta  
Email: [armagauthama@yahoo.com](mailto:armagauthama@yahoo.com)

### ABSTRACT

*Data is significant assets for a company to develop a strategy or to reach a sound decision. For the purpose, data can be processed using various means such as Operational database, Operational Application, or a data warehouse.*

*Data warehouse development phase Method is used in this study. This study is aimed to design and develop a data warehouse in STMIK AMIKOM Yogyakarta Library Unit. ETL process can be performed after the noise found is eliminated and the devised data warehouse is adjusted with the library needs. To conclude so, it is necessary to perform a data warehouse functional testing in which the report is analyzed.*

*Keywords: data warehouse, decision making, ETL, reporting*

### PENDAHULUAN

*Data warehouse* adalah suatu konsep dan kombinasi teknologi yang memfasilitasi organisasi untuk mengelola dan memelihara data historis yang diperoleh dari sistem atau aplikasi operasional [Ferdiana, 2008]. Pemakaian teknologi *data warehouse* hampir dibutuhkan oleh semua organisasi, tidak terkecuali Perpustakaan. *Data warehouse* memungkinkan integrasi berbagai macam jenis data dari berbagai macam aplikasi atau sistem. Hal ini menjamin mekanisme akses “satu pintu” bagi manajemen untuk memperoleh informasi, dan menganalisisnya untuk pengambilan keputusan.

Perpustakaan yang belum mempunyai *data warehouse* integrasi biasanya secara manual atau menggunakan software aplikasi yang digunakan untuk menghubungkan modul-modul yang terpisah. Pada dasarnya proses yang terjadi adalah aktifitas meringkas data yang telah tersimpan di *data warehouse* atau sistem basisdata (*database*) terkait.

Penelitian ini bertujuan untuk merancang dan membangun *data warehouse* pada Perpustakaan STMIK AMIKOM Yogyakarta. Hasil yang diharapkan adalah sebagai berikut.

1. Rancangan *star schema* untuk data warehouse Perpustakaan STMIK AMIKOM Yogyakarta.
2. Database *data warehouse*.
3. Proses loading per-periode.
4. Ringkasan data dari *data warehouse* yang diolah dengan *Reporting*

### Landasan Teori

*Data warehouse* adalah suatu paradigma baru dilingkungan pengambilan keputusan strategik. *Data warehouse* bukan suatu produk tetapi suatu lingkungan dimana user dapat menemukan informasi strategik [Poniah, 2001, hal:14]. *Data warehouse* adalah kumpulan data-data logik yang terpisah dengan database operasional dan merupakan suatu ringkasan. Adapun karakteristik dari *data warehouse* [Poniah, 2001,h.20-24] adalah sebagai berikut.

1. Berorientasi subyek

Tabel 1: Perbedaan database operasional dan *data warehouse*

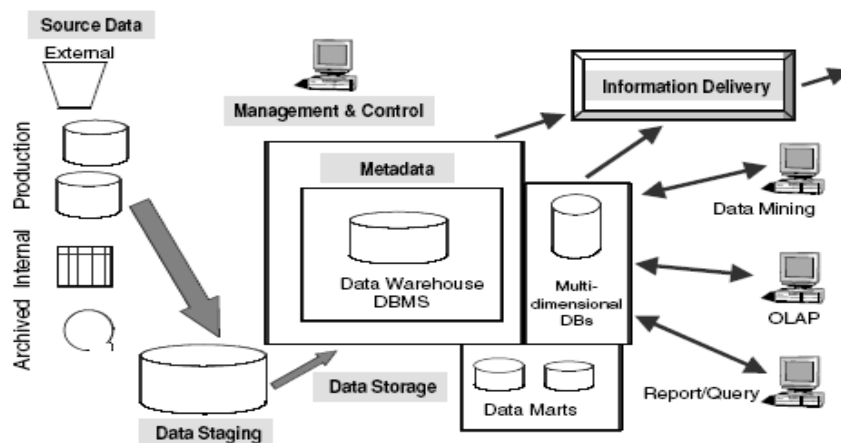
	Database operasional	<i>Data warehouse</i>
Isi data	Bernilai sekarang atau up-to-date	Arsip, <i>history</i> , rangkuman
Struktur data	Dioptimasi untuk transaksi, normalisasi	Dioptimasi untuk <i>query</i> yang kompleks, Unnormalisasi
Frekuensi akses	Tinggi	Sedang-rendah
Tipe akses	Read, update, delete	Read
Penggunaan	Update secara terus menerus	Update secara periodik
Users	Banyak	Lebih sedikit

Sumber: Poniah, 2001, hal: 11.

2. Data yang terintegrasi  
Sumber data yang ada dalam *data warehouse* tidak hanya berasal dari database operasional (*internal source*) tetapi juga berasal dari data diluar sistem (*external source*).
3. *Nonvolatile*  
Data dalam database operasional akan secara berkala atau periodik dipindahkan kedalam *data warehouse* sesuai dengan jadwal yang sudah ditentukan. Misal perhari,perminggu, perbulan, dan lain sebagainya.
4. *Time-Variant*  
Sistem operasional mengandung data yang bernilai sekarang sedangkan data dalam *data warehouse* mengandung data tidak hanya data terkini tetapi juga data *history* yang akan digunakan dalam analisis dan pengambilan keputusan. Waktu adalah dimensi penting yang harus didukung oleh semua *data warehouse*. Data untuk analisis dari berbagai sumber berisi berbagai nilai waktu, misalkan harian, mingguan, dan bulanan.
5. Ringkas  
Jika diperlukan, data operasional dikumpulkan ke dalam ringkasan-ringkasan.
6. *Granularity*  
Pada sistem operasional data dibuat secara *real-time* sehingga untuk mendapatkan informasi langsung dilakukan proses *query*.
7. Tidak ternormalisasi  
Data di dalam sebuah *data warehouse* biasanya tidak ternormalisasi dan sangat redundan.

Dasar dari suatu *data warehouse* adalah suatu data yang besar yang mengandung informasi bisnis. Data-data yang ada di dalam *data warehouse* bisa berasal dari banyak sumber, misalkan dari database operasional atau transaksional dan sumber dari luar misalkan dari web, penyedia jasa informasi, dari perusahaan lain, dan lain sebagainya. *Data warehouse* mengandung beberapa elemen penting antara lain [Mallach, 2000,h.473]:

1. Sumber data yang digunakan oleh *data warehouse*, database transaksional dan sumber data eksternal.
2. Proses ETL (*Extraction, Transformation, Loading*) dari sumber data ke database *data warehouse*.
3. Membuat suatu ringkasan atau *summary* terhadap *data warehouse* misalkan dengan menggunakan fungsi agregat.
4. Metadata.  
Metadata mengacu data tentang data. Metadata menguraikan struktur dan beberapa arti tentang data, dengan demikian mendukung penggunaan efektif atau tidak efektif dari data.
5. Database *data warehouse*.  
Database ini berisi data yang detail dan ringkasan data dari data yang ada di dalam *data warehouse*. Karena *data warehouse* tidak digunakan dalam proses transaksi individu, maka databasenya tidak perlu diorganisasikan untuk akses transaksi dan untuk pengambilan data, melainkan dioptimisasikan untuk pola akses yang berbeda di dalam analisis.
6. *Query Tools* yaitu dengan OLAP (*Online Analytical Processing*) dan data mining. Tool untuk *query* ini meliputi antarmuka pengguna akhir dalam mengajukan pertanyaan kepada database, dimana proses ini disebut sebagai *On-line Analytical Processing* (OLAP). Tool ini juga terdiri dari tool otomatis yang menemukan pola-pola di dalam data, yang sering disebut sebagai *data mining*. *Data warehouse* harus memiliki salah satu dari kedua tipe ini atau malah kedua-duanya.
7. User.  
Pengguna yang memanfaatkan *data warehouse* tersebut.



Gambar 1: Arsitektur Data Warehouse (Sumber: Poniah, 2001, hal: 29)

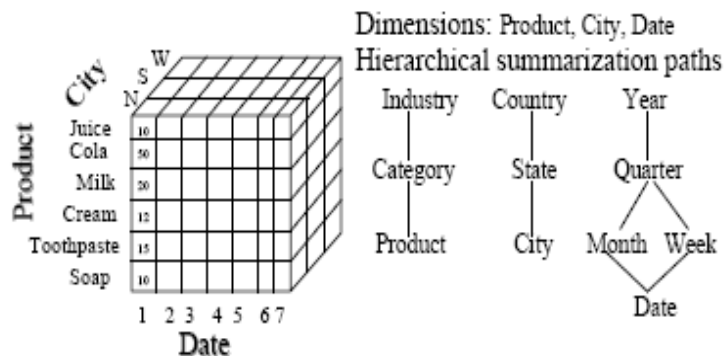
Pada sistem OLTP (*Online Transactional Processing*) digunakan suatu teknik pemodelan data yang disebut sebagai E-R (*Entity-Relationship*). Pada *data warehouse* digunakan teknik pemodelan data yang disebut *dimensional modelling technique*. Pemodelan dimensional adalah suatu model berbasis pemanggilan yang mendukung akses *query* volume tinggi. *Star Schema* adalah alat dimana pemodelan dimensional diterapkan dan berisi sebuah tabel fakta pusat. Tabel fakta berisi atribut deskriptif yang digunakan untuk proses *query* dan *foreign key* untuk menghubungkan ke tabel dimensi. Atribut analisis keputusan terdiri dari ukuran performa, metrik operasional, ukuran agregat, dan semua metrik yang lain yang diperlukan untuk menganalisis performa organisasi. Tabel fakta menunjukkan apa yang didukung oleh *data warehouse* untuk analisis keputusan. Tabel dimensi mengelilingi tabel fakta pusat. Tabel dimensi berisi atribut yang menguraikan data yang dimasukkan dalam tabel fakta.

Tabel dimensi waktu adalah suatu tabel dimensi yang harus ada dalam sebuah *data warehouse* karena setiap *data warehouse* adalah *time series*. Waktu merupakan dimensi pertama yang harus diperhatikan dalam proses *sort order* dalam suatu database karena ketika hal itu pertama kali dilakukan, *loading* data secara berturut-turut dalam interval waktu tertentu akan masuk kedalam tempat yang sebenarnya dalam sebuah *disk*.

Menurut Rainardi, 2008, ETL adalah suatu proses mengambil dan mengirim data dari data sumber ke *data warehouse*. Dalam proses pengambilan data, data harus bersih agar didapat kualitas data yang baik. Contohnya ada nomor telepon yang invalid, ada kode buku yang tidak eksis lagi, ada beberapa data yang *null*, dan lain sebagainya. Pendekatan tradisional pada proses ETL mengambil data dari data sumber, meletakkan pada *staging area*, dan kemudian mentransform dan meng-load ke *data warehouse*.

Model konseptual populer yang digunakan untuk *front-end tools*, desain database, dan *query* untuk OLAP adalah pandangan *multidimensional* dalam sebuah *data warehouse*. Dalam model data *multidimensional* berisi satu set ukuran numerik yang digunakan untuk obyek analisis. Contoh penjualan, anggaran, pendapatan, persediaan, ROI (*return on investment*), dan lain sebagainya [Chaudhuri].

Setiap ukuran numerik terikat dalam suatu set dimensi. Contohnya dimension penjualan terdiri dari nama kota, nama produk, dan tanggal ketika penjualan dibuat. Pandangan ukuran *multidimensional data* sebagai nilai dalam bagian *multidimension* atau *cube*. Setiap dimensi menjelaskan satu set atribut. Contohnya, dimensi produk terdiri atas empat atribut, yaitu kategori, industri, tahun produksi, dan rata-rata keuntungan. Dari gambar 2. dapat dilihat nama produk cola mempunyai kategori minuman, termasuk dalam industri makanan atau *food industry*, dan mempunyai keuntungan rata-rata 50%. Atribut dari suatu dimensi dapat digambarkan dengan suatu hubungan hirarki.



Gambar 2: Multidimensional data (Sumber: Chaudhuri)

Dalam proses analisis terhadap *data warehouse*, terdapat tiga kategori aktivitas yang dilakukan, yaitu:

1. *Reporting*
2. OLAP
3. *Data Mining*

### Reporting

Dalam konteks *data warehouse*, *report* adalah program yang menerima data dari *data warehouse* dan mempresentasikan ke user. *Report* dibangun dengan fungsi tertentu. Informasi yang diberikan disesuaikan dengan kepentingan user.

### OLAP (Online Analytic Processing)

Istilah *Online Analytic Processing* (OLAP) mengacu kepada berbagai aktivitas yang umumnya dilakukan oleh pengguna akhir didalam sistem online. Umumnya OLAP meliputi aktivitas-aktivitas seperti pembangkitan *query*, permintaan laporan khusus dan grafik, dan melakukan analisis statistik. OLAP menyediakan kemampuan pemodelan dan visualisasi untuk kumpulan data besar yang diambil dari database operasional dan lebih seringnya diambil dari *data warehouse*.

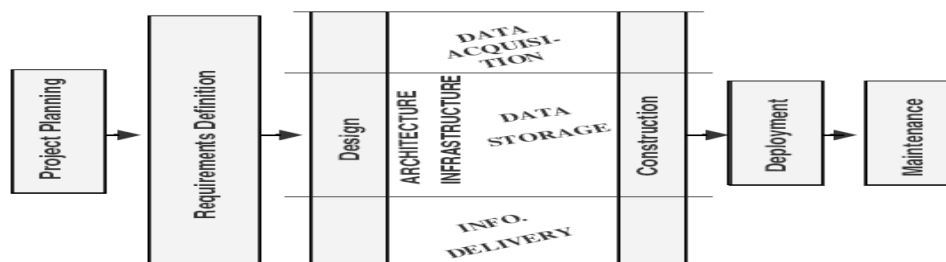
OLAP berbeda dengan *data mining* dalam hal para pengguna dapat mengajukan pertanyaan spesifik dan terbuka. Para pengguna yang pada umumnya analisis menjalankan sistem OLAP. Analisis mengarahkan OLAP, sedangkan *data mining* mencari hubungan dengan beberapa petunjuk dari analisis.

### Data Mining

*Data mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. *Data mining* adalah suatu proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine-learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar [Turban, 2005, hal:339].

### Fase pengembangan data warehouse

Proses pengembangan *data warehouse* terdiri dari beberapa fase, diantaranya fase perencanaan proyek, mendefinisikan kebutuhan, desain, konstruksi, *deployment*, dan *maintenance*. Gambar 3. menggambarkan proses dari pengembangan *data warehouse*.



Gambar 3: Fase Pengembangan Data Warehouse (Sumber: Poniah, 2001, hal: 74)

### Metode testing pada *data warehouse*

Setelah *data warehouse* dibangun menggunakan proses ETL dan menganalisisnya, langkah selanjutnya adalah melakukan pengujian terhadap *data warehouse*. Adapun metode pengujian terhadap *data warehouse* adalah sebagai berikut [Rainardi, 2008, hal:477]

1. *ETL Testing*  
Proses testing ini menyakinkan apakah perubahan terhadap sumber data juga merubah data dalam *data warehouse*. Dalam sistem *data warehouse* ada tiga bagian yang sangat pokok, yaitu ETL, *data store*, dan aplikasi. Proses ETL bisa memakan waktu 60%-80% dari proses pembangunan *data warehouse* [Rainardi, 2008, hal:478].
2. *Functional testing*  
Untuk menyakinkan apakah *data warehouse* yang dibangun sudah sesuai dengan *business requirement*.
3. *Performance testing*  
Digunakan untuk memastikan apakah *data warehouse* dapat menangani proses *load* dari sumber data.
4. *Security testing*  
Untuk memastikan bahwa user yang mengakses *data warehouse* adalah user yang memang mempunyai hak akses.
5. *User acceptance testing*  
Dalam *testing* ini, *user-end* menggunakan *data warehouse* untuk memverifikasi tingkat kegunaan dari *data warehouse*.
6. *End-to-end testing*  
Sistem diimplementasi dalam beberapa hari, selanjutnya di evaluasi.

Adapun jalan penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut.

1. Menentukan subyek *data warehouse*.  
Salah satu karakteristik dari *data warehouse* adalah *subject-oriented* sehingga langkah awal dalam membuat *data warehouse* adalah menentukan subyeknya.
2. Mendefinisikan kebutuhan  
Mendefinisikan kebutuhan terhadap sumber data yang dibutuhkan oleh *data warehouse* dan informasi yang ingin didapat dari *data warehouse* untuk mendukung keputusan manajemen.
3. Membuat rancangan *star schema* dan mendesain proses ETL.  
*Data warehouse* menggunakan model data dimensional atau sering disebut sebagai *star schema*. *Star schema* mempunyai dua bagian, yaitu tabel fakta dan tabel dimensi. Setelah *star schema* dibuat maka proses selanjutnya adalah melakukan proses ETL (*Extract, Transformation, Loading*) dari database operasional ke database *data warehouse*. Sebelum proses ETL ini diproses harus dibuat dulu desain untuk proses ETL-nya.
4. Melakukan proses ETL dari database operasional ke database *data warehouse*.  
Proses ETL secara periodik mengekstrak data dari sistem sumber, mentransformasikannya ke sebuah format yang umum, dan kemudian memuatnya ke dalam *data store* target, yang umumnya sebuah *data warehouse* atau *data mart*. ETL sangat penting untuk integrasi data dan *data warehousing*.
5. Membuat *cube* dan dimensional untuk *data warehouse*.
6. Menganalisis *data warehouse* menggunakan *reporting*.
7. Melakukan test pada *data warehouse*.

## HASIL DAN PEMBAHASAN

### Perencanaan Proyek

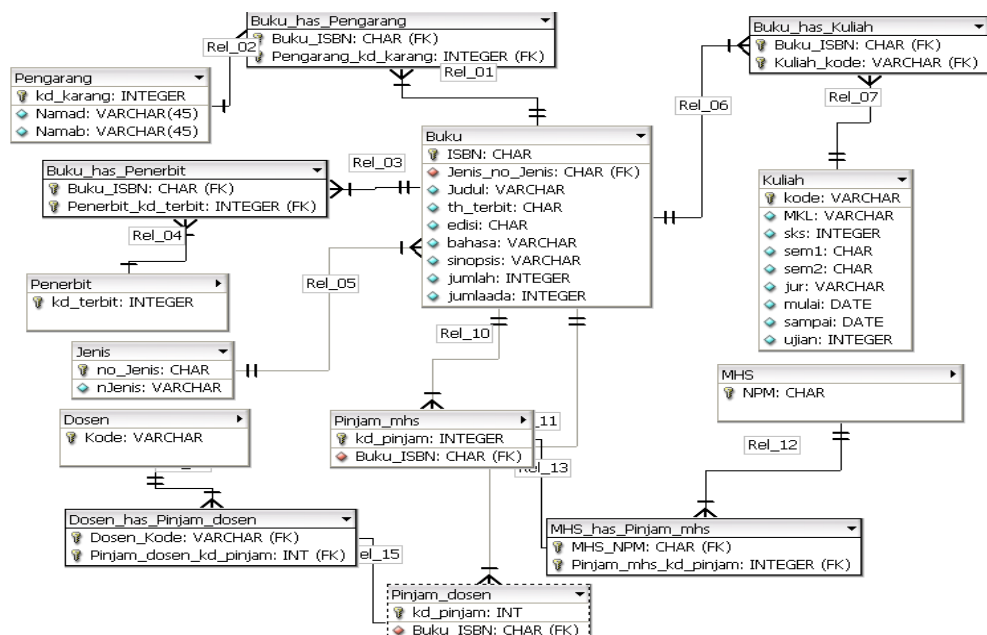
Pada fase perencanaan proyek dihasilkan subyek dari *data warehouse*, yaitu sirkulasi.

### Mendefinisikan Kebutuhan

Pada fase ini mendefinisikan semua sumber data yang digunakan dalam *data warehouse* dan mendefinisikan kebutuhan informasi yang harus dihasilkan oleh *data warehouse* melalui analisis *reporting*.

### Sumber data

Sumber data yang digunakan dalam *data warehouse* adalah data-data dalam database perpustakaan.



Gambar 4: ER-Diagram Perpustakaan

### Kebutuhan Informasi

Berdasarkan hasil pengamatan dan wawancara langsung kepada petugas atau manajemen perpustakaan didapat kebutuhan informasi sebagai berikut.

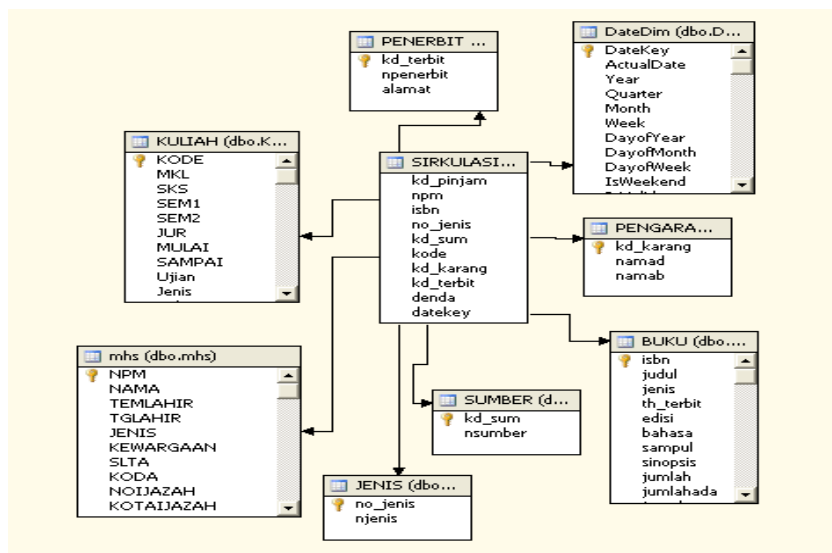
1. Tren terhadap buku-buku yang dipinjam oleh dosen dan mahasiswa.
2. Informasi mengenai jenis buku yang sering dipinjam.
3. Tren proses sirkulasi.
4. Informasi mengenai nama pengarang yang bukunya sering dipinjam.
5. Informasi mengenai nama penerbit yang bukunya sering dipinjam.

### Desain

Pada fase ini aktifitas yang dilakukan adalah membuat model data dimensional yang berupa *star Schema*, mendesain proses ETL (*Extraction, Transformation, Loading*), dan menganalisis metadata yang digunakan dalam *data warehouse*.

### Model data dimensional

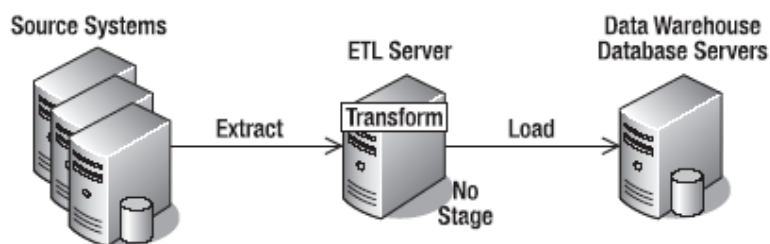
Berdasarkan ER-Diagram Perpustakaan dan kebutuhan informasi yang dibutuhkan oleh petugas dan manajemen perpustakaan maka model data dimensional yang dibuat dalam bentuk *star schema* untuk *data warehouse*.



Gambar 5: Star Schema untuk data warehouse perpustakaan

### Desain proses ETL

Proses selanjutnya setelah *star schema* dibuat adalah membuat desain proses ETL. Desain proses ETL yang dipakai bisa dilihat pada gambar 6 dibawah ini. Proses ETL mengambil data dari *source systems* menggunakan *query*. ETL berkoneksi dengan *source system* database dan mengambil data dengan *query*. Setelah data hasil *query* diambil langkah selanjutnya dilakukan eksekusi proses ETL dan mengirimnya ke database *data warehouse*.



Gambar 6: Desain proses ETL

### Konstruksi

Pada fase ini ada beberapa hal yang dilakukan, antara lain:

- 1) Proses ETL dari database perpustakaan ke database *data warehouse*.
  - a. Proses menghilangkan *noise* pada database sumber.
  - b. Proses mempopulasikan tabel dimensi .
  - c. Proses mempopulasi tabel *date dimension*.
  - d. Proses mempopulasikan tabel fakta.
- 2) Membuat *Cube* dan dimensional pada *data warehouse*.
- 3) Menganalisis data warehouse menggunakan *reporting*.
- 4) Melakukan test pada *data warehouse*.

### Proses menghilangkan noise pada database sumber

Proses ETL dari database perpustakaan ke database *data warehouse*. Sebelum proses ETL dilakukan perlu adanya proses menghilangkan *noise* yang ada pada database sumber atau database OLTP. Setelah dianalisis ternyata *noise* ada pada tabel pinjam\_mhs, yaitu adanya data yang bernilai *null* pada kolom kd\_pinjam\_mhs di tabel pinjam\_mhs.

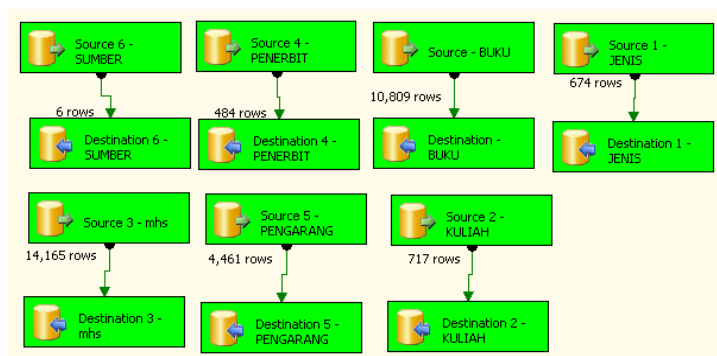
```
select * from sirkulasi left outer join pinjam_mhs on
sirkulasi.kd_pinjam_mhs=pinjam_mhs.kd_pinjam_mhs
```

kd_pinjam_mhs	nim	tgl_pinjam	tgl_kembali	kd_pinjam_mhs	kd_buku
69	01.02.3754	2003-09-13 00:00:00.000	2003-09-13 00:00:00.000	NULL	NULL
86174	03.01.1560	2005-09-17 00:00:00.000	2005-09-17 00:00:00.000	NULL	NULL
199778	06.02.6279	2008-05-17 00:00:00.000	NULL	NULL	NULL
161	02.02.4423	2003-09-15 00:00:00.000	2003-09-15 00:00:00.000	NULL	NULL
78685	04.01.1812	2005-07-06 00:00:00.000	2005-07-06 00:00:00.000	NULL	NULL
151	01.02.3779	2003-09-15 00:00:00.000	2003-09-15 00:00:00.000	NULL	NULL
88277	04.22.0342	2005-09-28 00:00:00.000	2005-09-28 00:00:00.000	NULL	NULL
87073	05.11.0949	2005-09-21 00:00:00.000	2005-09-21 00:00:00.000	NULL	NULL
196809	06.12.2026	2008-04-22 00:00:00.000	NULL	NULL	NULL
196731	07.01.2187	2008-04-22 00:00:00.000	NULL	NULL	NULL
189938	07.12.2334	2008-03-01 00:00:00.000	NULL	NULL	NULL
42	01.02.3498	2003-09-12 00:00:00.000	2003-09-12 00:00:00.000	NULL	NULL
174545	06.02.6386	2007-10-23 00:00:00.000	NULL	NULL	NULL
72334	04.11.0603	2005-05-17 00:00:00.000	2005-05-17 00:00:00.000	NULL	NULL
184719	07.22.0766	2008-01-08 00:00:00.000	NULL	NULL	NULL
86538	04.22.0384	2005-09-19 00:00:00.000	2005-09-19 00:00:00.000	NULL	NULL
184772	05.01.2010	2008-01-09 00:00:00.000	NULL	NULL	NULL
78981	03.01.1617	2005-07-11 00:00:00.000	2005-07-11 00:00:00.000	NULL	NULL
60371	04.01.1773	2005-03-08 00:00:00.000	2005-03-08 00:00:00.000	NULL	NULL

Gambar 7: Hasil *query* yang menunjukkan adanya *noise*

### Proses mempopulasikan tabel dimensi dan tabel fakta

Untuk mengekstrak data dari database OLTP ke database *data warehouse* diperlukan suatu teknik populasi. Teknik populasi ini terdiri dari teknik untuk mempopulasikan tabel dimensi dan teknik untuk mempopulasikan tabel fakta. Mempopulasikan tabel dimensi untuk tabel buku, tabel pengarang, tabel penerbit, tabel sumber, tabel jenis, tabel mhs, tabel kuliah dilakukan secara langsung dari database OLTP. Proses mempopulasi tabel dimensi dan tabel fakta menggunakan SSIS (SQL Server Integration Service), yaitu suatu tool yang dimiliki oleh SQL Server 2005 yang digunakan untuk membantu proses ETL.



Gambar 8: Hasil dari proses eksekusi untuk tabel dimensi

Proses selanjutnya adalah proses *loading* berkala pada setiap tabel dimension. Proses *loading* berkala ini menggunakan *transact SQL* yang dilakukan pada database *data warehouse*, yaitu perpustakaan dw.

- a. Proses *loading* berkala pada tabel pengarang

```
--jika akan melakukan penambahan tinggal mengubah kd_karang-nya saja.
insert into pengarang
select * from [perpustakaan].dbo.pengarang where kd_karang>=3000
```

- b. Proses *loading* berkala pada tabel penerbit

```
--jika akan melakukan penambahan tinggal mengubah kd_terbit-nya saja.
insert into penerbit
select * from [perpustakaan].dbo.penerbit where kd_terbit<=300
```

- c. Proses *loading* berkala pada tabel jenis

```
insert into jenis
select * from [perpustakaan].dbo.jenis where
[perpustakaan].dbo.jenis.no_jenis<>jenis.no_jenis
```



- d. Proses loading berkala pada tabel buku

```
insert into buku
select * from [perpustakaan].dbo.buku where
[perpustakaan].dbo.buku.isbn<>buku.isbn
```

- e. Proses loading berkala pada tabel kuliah

```
insert into kuliah
select * from [perpustakaan].dbo.kuliah where
[perpustakaan].dbo.kuliah.kode<>kuliah.kode
```

- f. Proses loading berkala pada tabel mhs

```
insert into mhs
select * from [perpustakaan].dbo.mhs where
[perpustakaan].dbo.mhs.npm<>mhs.npm
```

- g. Proses loading berkala pada tabel buku

```
insert into sumber
select * from [perpustakaan].dbo.sumber where kd_sum>6
```

#### Proses mempopulasikan tabel dimensi datedim

Setiap *data warehouse* harus mempunyai tabel dimensi datedim. Pada gambar dibawah ini adalah proses pembuatan tabel dimensi datedim pada database *data warehouse*.

```
CREATE TABLE [dbo].[DateDim] (
    [DateKey] [int] IDENTITY (1, 1) NOT NULL ,
    [ActualDate] [datetime] NOT NULL ,
    [Year] [int] NOT NULL ,
    [Quarter] [int] NOT NULL ,
    [Month] [int] NOT NULL ,
    [Week] [int] NOT NULL ,
    [DayofYear] [int] NOT NULL ,
    [DayofMonth] [int] NOT NULL ,
    [DayofWeek] [int] NOT NULL ,
    [IsWeekend] [bit] NOT NULL ,
    [IsHoliday] [bit] NOT NULL ,
    [Comments] [varchar] (20) COLLATE Latin1_General_CI_AI NULL ,
    [CalendarWeek] [int] NOT NULL ,
    [BusinessYearWeek] [int] NOT NULL ,
    [LeapYear] [tinyint] NOT NULL
) ON [PRIMARY]
```

Setelah tabel datedim dibuat, langkah selanjutnya membuat suatu procedure yang digunakan untuk mekanisme penginputan data berdasarkan periode tertentu.

```
CREATE PROCEDURE sp_createTimeDim AS
--menghapus table timedim
TRUNCATE TABLE DateDim
--mendeklarasikan variabel
DECLARE @DT DATETIME
DECLARE @YEAR INT
DECLARE @QUARTER INT
DECLARE @MONTH INT
DECLARE @WEEK INT
DECLARE @DayofYear INT
DECLARE @DayofMonth INT
DECLARE @DayofWeek INT
DECLARE @IsWeekend BIT
DECLARE @IsHoliday BIT
DECLARE @CalendarWeek INT
DECLARE @DayName VARCHAR(20)
DECLARE @MonthName VARCHAR(20)
DECLARE @BusinessYearWeek INT
DECLARE @LeapYear BIT
```

```

--inisialisasi variabel
SELECT @BusinessYearWeek =0
SELECT @CalendarWeek = 1
SELECT @LeapYear =0
--Tanggal awal pada table datedim
SELECT @DT = '1/1/2000'
--perulangan dimulai sampai tanggal terakhir pada table datedim
WHILE (@DT <= '1/31/2020')
BEGIN
    SELECT @IsWeekend =0
    SELECT @YEAR = DATEPART (YEAR, @DT)
    SELECT @QUARTER = DATEPART (QUARTER, @DT)
    SELECT @MONTH = DATEPART (MONTH , @DT)
    SELECT @WEEK = DATEPART (WEEK , @DT)
    SELECT @DayofYear = DATEPART (DY , @DT)
    SELECT @DayofMonth = DATEPART (DAY , @DT)
    SELECT @DayofWeek = DATEPART (DW , @DT)
    --note if weekend or not
    IF ( @DayofWeek = 1 OR @DayofWeek = 7 )
    BEGIN
        SELECT @IsWeekend = 1
    END
    --add 1 every time we start a new week
    IF ( @DayofWeek = 1)
    BEGIN
        SELECT @CalendarWeek = @CalendarWeek +1
    END

    --add business rule (need to know complete weeks in a year, so a partial week in new year set to 0)
    IF ( @DayofWeek != 1 AND @DayofYear = 1)
    BEGIN
        SELECT @BusinessYearWeek = 0
    END
    IF ( @DayofWeek = 1)
    BEGIN
        SELECT @BusinessYearWeek = @BusinessYearWeek +1
    END
    --add business rule (start counting business weeks with first complete week)
    IF (@BusinessYearWeek =53)
    BEGIN
        SELECT @BusinessYearWeek = 1
    END
    --check for leap year
    IF ((@YEAR % 4 = 0) AND (@YEAR % 100 != 0 OR @YEAR % 400 = 0))
        SELECT @LeapYear =1
    ELSE SELECT @LeapYear =0

    --insert values into table Datedim
    INSERT DateDim (ActualDate, Year, Quarter, Month, Week, DayofYear, DayofMonth, DayofWeek,
    IsWeekend, CalendarWeek, BusinessYearWeek, LeapYear)
    VALUES (@DT, @YEAR, @QUARTER, @MONTH, @WEEK, @DayofYear, @DayofMonth,
    @DayofWeek, @IsWeekend, @CalendarWeek, @BusinessYearWeek, @LeapYear)
    --increment the date one day
    SELECT @DT = DATEADD(DAY, 1, @DT)
END
GO

```

Penambahan data perperiode ke tabel timedim melalui bagian script diatas yaitu pada bagian dibawah ini.

```

--Tanggal awal pada table datedim
SELECT @DT = '1/1/2000'
--perulangan dimulai sampai tanggal terakhir pada table datedim
WHILE (@DT <= '1/31/2020')

```

Tanggal terakhir yang diinputkan bisa dilihat pada akhir perulangan yaitu kurang dari tanggal 31 Januari 2020. Jika ingin dilakukan proses penginputan kembali perubahannya pada select @DT='1/31/2020' dan diakhiri pada tanggal yang diinginkan.

### Proses mempopulasikan tabel fakta

Tabel fakta dibuat dari hasil *query* beberapa tabel pada database sumber. *Query* untuk membuat membuat tabel fakta bisa dilihat pada script dibawah ini.

```
select sirkulasi1.kd_pinjam,mhs.npm,buku.isbn,jenis.no_jenis,
sumber.kd_sum,kuliah.kode,
pengarang.kd_karang,penerbit.kd_terbit,
pinjam_mhs.denda,[perpustakaan dw].dbo.datedim.datekey
from pengarang join karang on
pengarang.kd_karang=karang.id_karang join buku
on buku.isbn=karang.isbn join terbit on buku.isbn=terbit.isbn join
penerbit on penerbit.kd_terbit=terbit.id_terbit join data_buku on
buku.isbn=data_buku.isbn join buku_mk on buku_mk.isbn=buku.isbn join
kuliah on kuliah.kode=buku_mk.kode_mk join jenis on jenis.no_jenis=
buku.jenis join sumber on sumber.kd_sum=data_buku.sumber join
pinjam_mhs on
pinjam_mhs.kd_buku=data_buku.kd_buku join sirkulasi1 on
pinjam_mhs.kd_pinjam_mhs=sirkulasi1.kd_pinjam join mhs on
mhs.npm=sirkulasi1.nim join [perpustakaan dw].dbo.datedim on
[perpustakaan dw].dbo.datedim.actualdate=sirkulasi1.tgl_pinjam
order by kd_pinjam
```

Setelah terbentuk tabel fakta yang bernama sirkulasi, langkah selanjutnya membuat mekanisme yang digunakan untuk melakukan penginputan data pada tabel sirkulasi pada periode tertentu.

### Analisis pada data warehouse

Analisis pada *data warehouse* pada penelitian ini dilakukan dengan cara membuat suatu *reporting* sesuai dengan *business requirement*. Adapun beberapa *report* yang telah dibuat adalah sebagai berikut:

1. Laporan jumlah peminjaman pertahun.

#### Peminjaman pertahun

Year	jml peminjam
2001	31
2002	20
2003	15127
2004	57042
2005	70851
2006	84559
2007	79681
2008	37734

Proses pembuatan laporan diatas bisa dilihat pada script dibawah ini.

```
SELECT DateDim.Year, COUNT(SIRKULASI.kd_pinjam) AS jml_peminjam
FROM SIRKULASI INNER JOIN
DateDim ON SIRKULASI.datekey = DateDim.DateKey
GROUP BY DateDim.Year
```

2. Laporan buku terlaris.

Adapun cara membuatnya bisa dilihat pada script dibawah ini.

```
SELECT BUKU.isbn, BUKU.judul, PENGARANG.namad + PENGARANG.namab AS Pengarang,
COUNT(BUKU.isbn) AS jml_dipinjam
FROM BUKU INNER JOIN
SIRKULASI ON BUKU.isbn = SIRKULASI.isbn INNER JOIN
PENGARANG ON SIRKULASI.kd_karang = PENGARANG.kd_karang
GROUP BY BUKU.isbn, BUKU.judul, PENGARANG.namad, PENGARANG.namab
ORDER BY jml_dipinjam DESC
```

## 3. Laporan hari teramai.

LAPORAN HARI TERAMAI

Hari ☺	Jumlah dipinjam
2	72504
3	72143
4	69164
5	59252
6	45406
7	26490
1	86

Adapun proses pembuatan laporan diatas bisa dilihat pada script dibawah ini.

```
SELECT DateDim.DayofWeek AS hari, COUNT(SIRKULASI.kd_pinjam) AS jml_pinjam
FROM SIRKULASI INNER JOIN
DateDim ON SIRKULASI.datekey = DateDim.DateKey
GROUP BY DateDim.DayofWeek
ORDER BY jml_pinjam DESC
```

## 4. Laporan pengarang terlaris.

LAPORAN PENGARANG TERLARIS

Nama Pengarang	Jumlah Dipinjam
wahana komputer	24514
MADCOMMADIUN	13917
jogiyanto HM.	11297
M. Suyanto	8954
abdulkadir	7131
-Andi Yogyakarta	6350
EmaUtami	4537
GregoriusAgung	4181
Edi SMulyanta	3939
-MADCOMS	3711
Onno WPurbo	3402
LukmanulHakim	3147
InsapSantosa	3122
ianchandra k	3077
SuryantoThabrani	2952

Adapun cara pembuatan laporan diatas bisa dilihat pada script dibawah ini.

```
SELECT PENGARANG.namad + PENGARANG.namab AS pengarang,
COUNT(SIRKULASI.kd_pinjam) AS jml_pinjam
FROM BUKU INNER JOIN
SIRKULASI ON BUKU.isbn = SIRKULASI.isbn INNER JOIN
PENGARANG ON SIRKULASI.kd_karang = PENGARANG.kd_karang
GROUP BY PENGARANG.namad, PENGARANG.namab
ORDER BY jml_pinjam DESC
```

## 5. Laporan penerbit terlaris.

PENERBIT TERLARIS

NAMA PENERBIT	JUMLAH DIPINJAM
andi offset	146567
elex media komputindo	83198
Gramedia	10511
Prenhallindo	8203
Informatika	8000
Gava Media	7654
Wahana Komputer	6433
Erlangga	6256
BPFE	6118
Salemba Infotek	5926
Mizan	4346
UPT Penerbitan AMIKOM	4040
Microsoft	3274
Binarupa Aksara	2866
Glasshaus	2415
Salemba Empat	2006

Adapun cara pembuatan laporan diatas bisa dilihat pada script dibawah ini.

```
SELECT PENERBIT.npenerbit AS nama_penerbit, COUNT(SIRKULASI.kd_pinjam) AS jml_pinjam
FROM SIRKULASI INNER JOIN
PENERBIT ON SIRKULASI.kd_terbit = PENERBIT.kd_terbit
GROUP BY PENERBIT.npenerbit
ORDER BY jml_pinjam DESC
```

### Testing terhadap data warehouse

Sama dengan sistem-sistem IT yang lain, dalam membangun *data warehouse* juga diperlukan proses *testing*. Berdasarkan metode *testing* yang ada dalam *data warehouse*, proses *testing* yang dilakukan pada penelitian ini adalah sebagai berikut.

#### 1. ETL Testing

*Testing* yang dilakukan pada proses ETL ini antara lain memastikan proses *loading* yang dilakukan per-periode sudah masuk ke tabel dan kolom yang sesuai, serta pada waktu yang tepat. Proses yang dilakukan adalah dengan mencocokkan apakah jumlah record hasil *query* pada database operasional sama dengan jumlah record yang dihasilkan dari proses ETL.

#### 2. Functional testing

*Functional testing* yang dilakukan yaitu dengan mengecek apakah *data warehouse* yang dibangun sudah sesuai dengan *business requirement*. Tahapan yang dilakukan yaitu dengan cara melakukan analisis pada *data warehouse* dengan metode *reporting* dan hasilnya dicocokkan dengan *business requirement* pada tahapan mendefinisikan kebutuhan.

### KESIMPULAN

Adapun kesimpulan yang bisa diambil dari penelitian ini adalah sebagai berikut:

1. *Noise* yang ditemukan pada proses ETL berupa adanya *field foreign key* yang bernilai *null*.
2. Pada proses mempopulasikan tabel fakta *noise* yang ditemukan dihilangkan terlebih dahulu sebelum proses ETL dapat dilanjutkan.
3. *Data warehouse* yang dibangun sudah dilengkapi dengan proses *loading* berkala dalam bentuk otomatisasi sehingga memudahkan pihak *administrator data warehouse*.
4. *Data warehouse* yang dirancang dan dibangun sudah disesuaikan dengan kebutuhan dari manajemen perpustakaan, dibuktikan dengan adanya *functional testing* pada *data warehouse* yaitu berupa analisis *reporting*.

### Saran

Adapun saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Pada saat fase perancangan proyek, subyek *data warehouse* bisa dikembangkan menjadi beberapa subyek sesuai dengan kebutuhan manajemen perpustakaan.
2. Diperlukan pengembangan bahwa *data warehouse* perpustakaan sebagai *data mart* dari suatu *data warehouse* di STMIK AMIKOM Yogyakarta.
3. Pada saat analisis terhadap *data warehouse* perlu dikembangkan alat analisis, yaitu OLAP (*Online Analytic Processing*) dan *data mining*.

### DAFTAR PUSTAKA

- Chaudhuri, Surajit, *An Overview of Data Warehousing and OLAP Technology* [Online] <http://www.sigmod.org/record/issues/9703/chaudhuri.ps>. tanggal akses 21 Mei 2008, pukul 11.23 WIB.
- Ferdiana, Ridi, 2008, *Strategi pengelolaan asset data menggunakan konsep data warehouse dalam cara pandang organisasi*. [Online] [http://micresearch.net/file.axd?file=Data+warehouse+dalam+definisi+organisasi+dan+bisnis.p](http://micresearch.net/file.axd?file=Data+warehouse+dalam+definisi+organisasi+dan+bisnis.pdf) df tanggal akses 21 Mei 2008, pukul 10.30 WIB
- Han, Jiawei; Kamber, Micheline. 2006. *Data Mining: Concepts and Techniques*. San Fransisco: Morgan Kaufmann.
- Hutabarat, Bernaridho I., *Data Warehouse dengan SQL Server 2005*. Elex Media Komputindo: Yogyakarta.
- Kalyani, 2008, *Data warehouse*. [Online] [http://www.cse.buffalo.edu/DBGROUP/nachi/ecopres/kalyani.ppt#258,3,OLTP vs. Warehousing](http://www.cse.buffalo.edu/DBGROUP/nachi/ecopres/kalyani.ppt#258,3,OLTP%20vs.%20Warehousing). Tanggal akses 4 September 2008, pukul 09.40 WIB.
- Kimball, Ralph; Caserta, Joe. 2004. *The Data Warehouse ETL Toolkit*. New Delhi: Wiley Publishing.
- Mallach, Efrem G., 2000. *Decision Support and Data Warehouse Systems*. Singapore: Irwin McGraw Hill.

- Nicholson, Scott. 2003. *The Bibliomining Process: Data Warehouse dan Data Mining for Library Decision-Making*. [Online] <http://bibliomining.org>. Tanggal akses 17 April 2008, pukul 09.00 WIB.
- Ponniah, Paulraj. 2001. *Data Warehouse Fundamentals: a Comprehensive Guide for IT Professional*. New York: John Wiley & Sons.
- Populate Time Dimension of AdventureWorksDW Sample Database and use it in your Datawarehouse/cube. [Online] <http://blogs.msdn.com/azazr/archive/2008/05/09/populate-time-dimension-of-adventureworksdw-sample-database-and-use-it-in-your-datawarehouse-cube.aspx>. Tanggal akses 22 Juli 2008, Pukul 10.45.
- Rainardi, Vincent, 2008, *Building a Data Warehouse with Examples in SQL Server*. Apress: New York.
- Tang, ZhaoHui; MacLennan, Jamie. 2005. *Data Mining with SQL Server 2005*. Indiana Polis: Wiley Publishing.  
Tanggal akses 21 April 2008, pukul 10.15 WIB.
- Turban, 2005, *Decision Support Systems and Intelligent Systems ( Sistem Pendukung Keputusan dan Sistem Cerdas ) Jilid 1*. Andi Offset: Yogyakarta.
- Winpec Solution, 2007, *Membuat Aplikasi Reporting Service dengan SQL Server 2005*. Elex Media Komputindo: Jakarta.
- Yuadi, Imam, 2007, *Perpustakaan Digital: Paradigma, Konsep, dan Teknologi Informasi yang digunakan*. [Online] <http://journal.unair.ac.id/filerPDF/PERPUSTAKAAN%20DIGITAL.pdf>