

BAB IV

ANALISIS DAN PERANCANGAN AWAL DATA WAREHOUSE

Proses Loading Data dari Sumber ke Staging Area I

4.1 Analisis

PT Sumber Buana Motor (SBM) merupakan perusahaan yang berperan dalam bidang pelayanan jasa penjualan, pemeliharaan, dan penyediaan suku cadang kendaraan bermotor. Perusahaan ini berperan sebagai *main dealer* untuk wilayah Yogyakarta, Magelang, Cilacap, Banyumas, dan Surakarta. Sepanjang tahun, perusahaan ini memberikan pelayanan bagi para konsumen yang akan melakukan transaksi pembelian kendaraan bermotor (menangani transaksi penjualan kendaraan bermotor dalam jumlah yang banyak). Perusahaan akan mencatat data konsumen dan data transaksi penjualan tersebut secara detail.

Seiring dengan kondisi perusahaan yang semakin berkembang, maka data-data yang dimiliki oleh perusahaan akan semakin menumpuk. Hal tersebut dapat dilihat dari data penjualan yang semakin meningkat tiap tahunnya, dimana terdapat ratusan atau ribuan transaksi yang dilakukan oleh konsumen setiap bulannya. Dari data-data tersebut tentunya akan lebih berguna apabila dapat menghasilkan suatu informasi. Akan tetapi, pihak perusahaan belum dapat menggali informasi yang tersembunyi dalam kumpulan data tersebut secara efektif sehingga tidak cukup membantu dalam proses pengambilan keputusan yang strategis.

Untuk maksud itulah, maka dibuat suatu *data warehouse* yang nantinya akan lebih memudahkan

perusahaan dalam pembuatan laporan-laporan serta membantu proses-proses bisnis yang membutuhkan informasi secara cepat dan akurat sehingga dapat menekan biaya yang harus dikeluarkan oleh perusahaan dari aktifitas-aktifitas tersebut.

Dengan data warehouse ini, pihak perusahaan dapat memperoleh informasi-informasi yang penting sehingga membantu perusahaan untuk mengetahui:

1. Berapa jumlah penjualan tunai kendaraan bermotor yang didapatkan oleh masing-masing sales pada periode 2001-2006?
2. Berapa jumlah penjualan kredit kendaraan bermotor yang didapatkan oleh masing-masing sales pada periode 2001-2006?
3. Berapa jumlah penjualan tunai berdasarkan tipe dan model kendaraan bermotor yang diperoleh masing-masing outlet pada periode 2001-2006?
4. Berapa jumlah penjualan kredit berdasarkan tipe dan model kendaraan bermotor yang diperoleh masing-masing outlet pada periode 2001-2006?
5. Berapa jumlah total penjualan kendaraan bermotor yang diperoleh masing-masing outlet pada periode 2001-2006?
6. Berapa jumlah pelanggan SBM berdasarkan kabupaten asalnya?
7. Berapa jumlah pendapatan yang diperoleh dari penjualan tunai kendaraan bermotor di tiap outlet pada periode 2001-2006?
8. Berapa jumlah piutang kredit yang diberikan SBM kepada pelanggannya di tiap outlet pada periode 2001-2006?

9. Berapa jumlah pendapatan yang didapatkan dari pembayaran angsuran pokok per tipe kendaraan pada periode 2001-2006?
10. Berapa jumlah pendapatan yang didapatkan dari pembayaran bunga angsuran per tipe kendaraan pada periode 2001-2006?
11. Berapa jumlah pendapatan yang diterima SBM dari pembayaran uang muka setiap bulannya selama periode 2001-2006?
12. Berapa nilai total angsuran pokok dan bunga angsuran beserta jumlah cicilan angsuran yang dibayarkan oleh pelanggan per outlet untuk periode waktu tertentu?
13. Outlet mana sajakah yang termasuk dalam peringkat tiga besar dalam perolehan pendapatan dari penjualan tunai per wilayah pada tahun tertentu?
14. Berapa keuntungan yang diperoleh perusahaan dari penjualan kredit per produk pada tahun tertentu?
15. Tiga peringkat besar dalam perolehan total penjualan masing-masing sales per outlet pada tahun tertentu?

4.2 Perancangan Data Warehouse

Tahap awal dari pembangunan data warehouse adalah membuat rancangan dari data warehouse yang akan dibuat. Gambar berikut menunjukkan pemodelan dimensional berdasarkan analisis yang telah dilakukan pada data sumber:

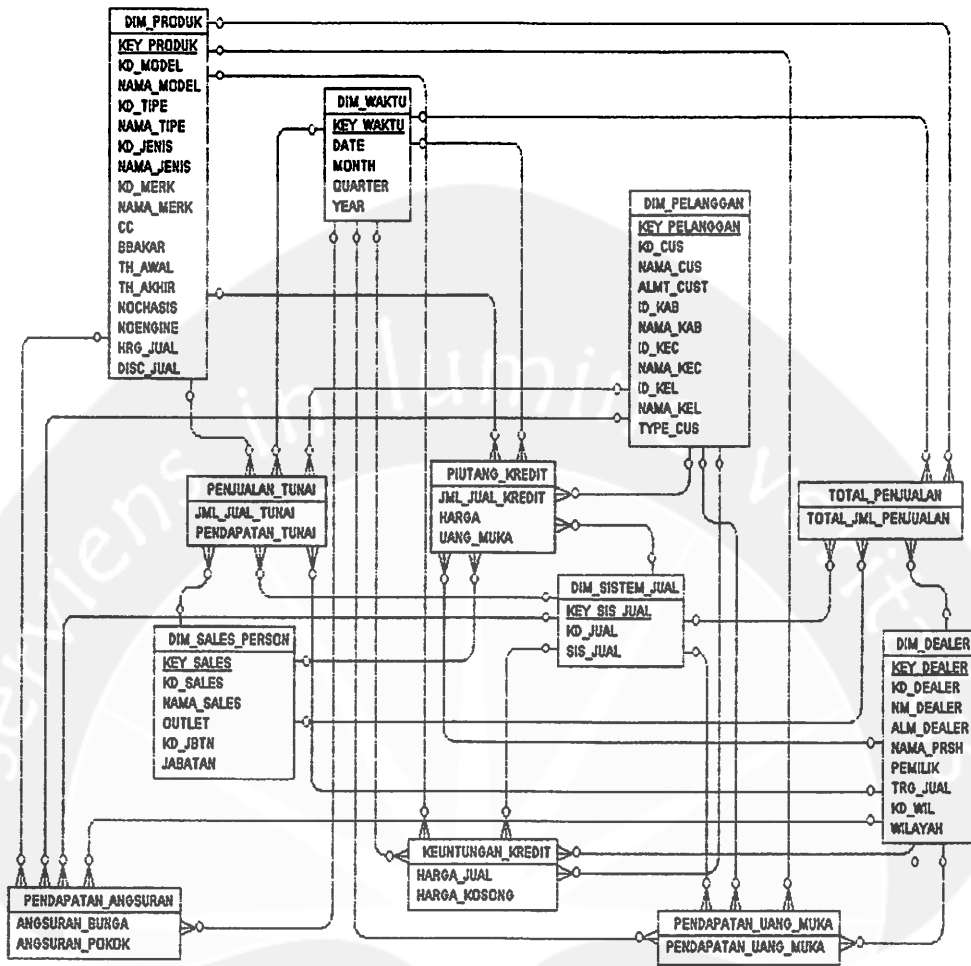
Information Subject : Penjualan

Dimensi

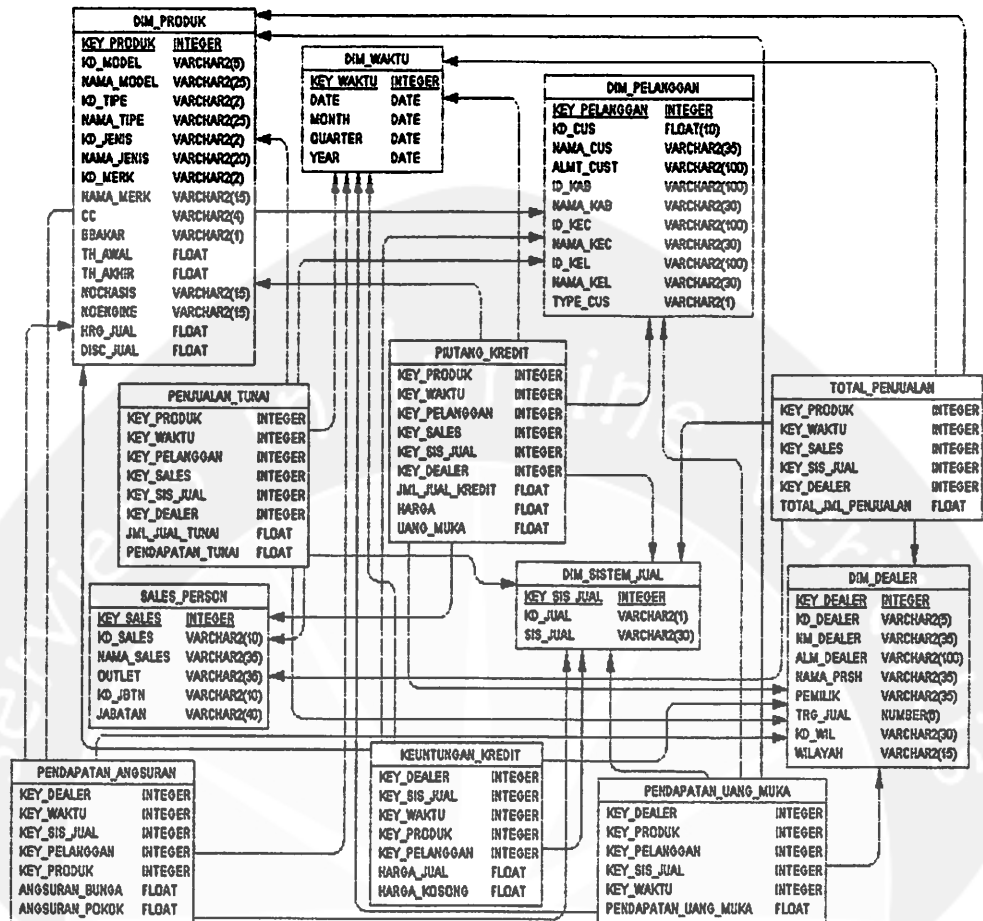
WAKTU	SALES PERSON	SISTEM JUAL	DEALER	PELANGGAN	PRODUK
Tahun	Kategori	Sistem Jual	Wilayah	Nama Kabupaten	Nama Merk
Kuartal	Nama Sales		Nama Dealer	Nama Kecamatan	Nama Jenis
Bulan	Outlet		Nama Perusahaan	Nama Kelurahan	Nama Tipe
Tanggal			Alamat Dealer	Nama Pelanggan	Nama Model
			Pemilik	Alamat Pelanggan	Bbakar
			Target Jual	Tipe Pelanggan	CC
					Diskon Jual
					Harga Jual
					Nomor Chasis
					Nomor Engine
					Tahun Akhir
					Tahun Awal

Fakta : Penjualan Tunai, Penjualan Kredit, Total Penjualan, Pendapatan Tunai, Piutang Kredit, Keuntungan Kredit, Pendapatan Angsuran, Pendapatan Uang Muka

Gambar 4.1 Information Package Diagram : Penjualan



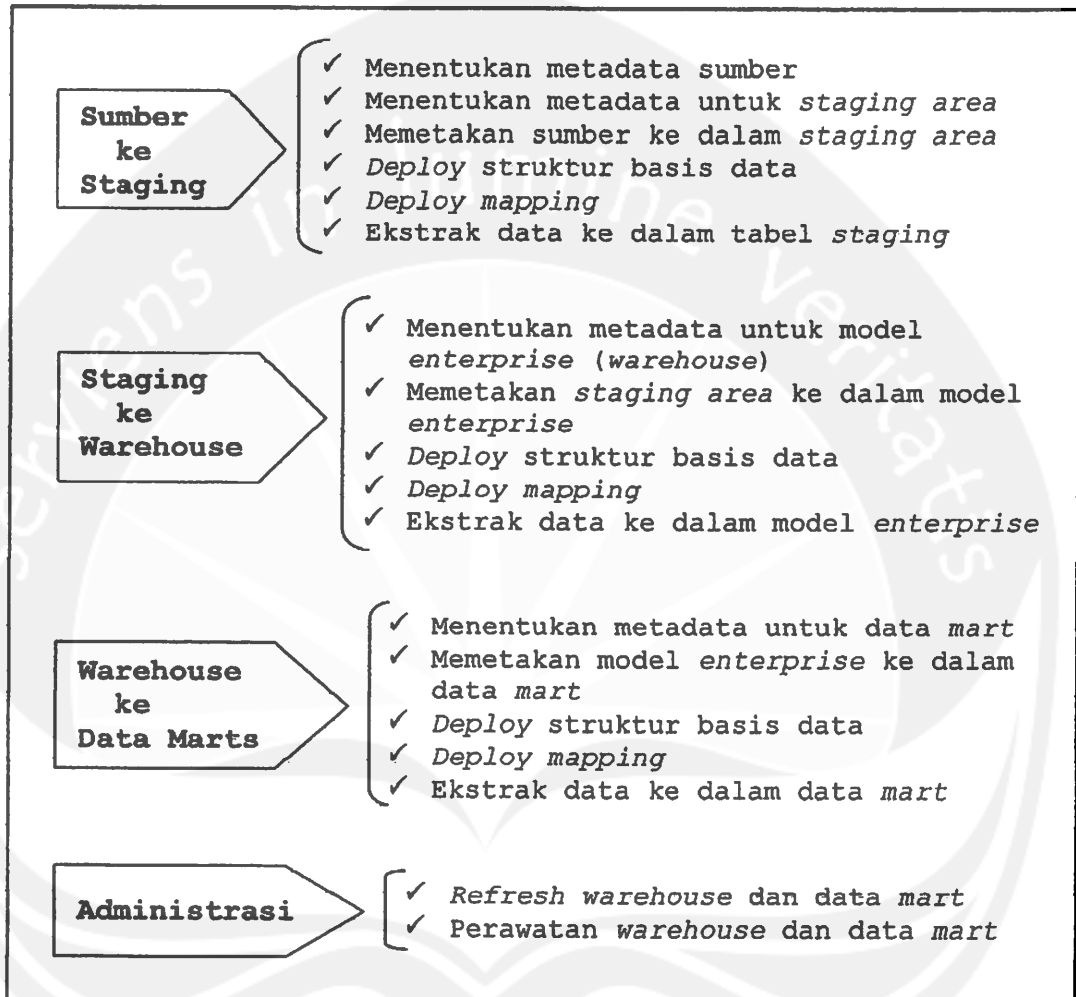
Gambar 4.2 Star Schema : Penjualan



Gambar 4.3 Star Schema: Penjualan

4.3 Proses Pembangunan Data Warehouse

Dalam pembangunan suatu data warehouse, ada beberapa tahapan yang harus dilakukan. Berikut ini adalah tahapan-tahapan pembangunan data warehouse:



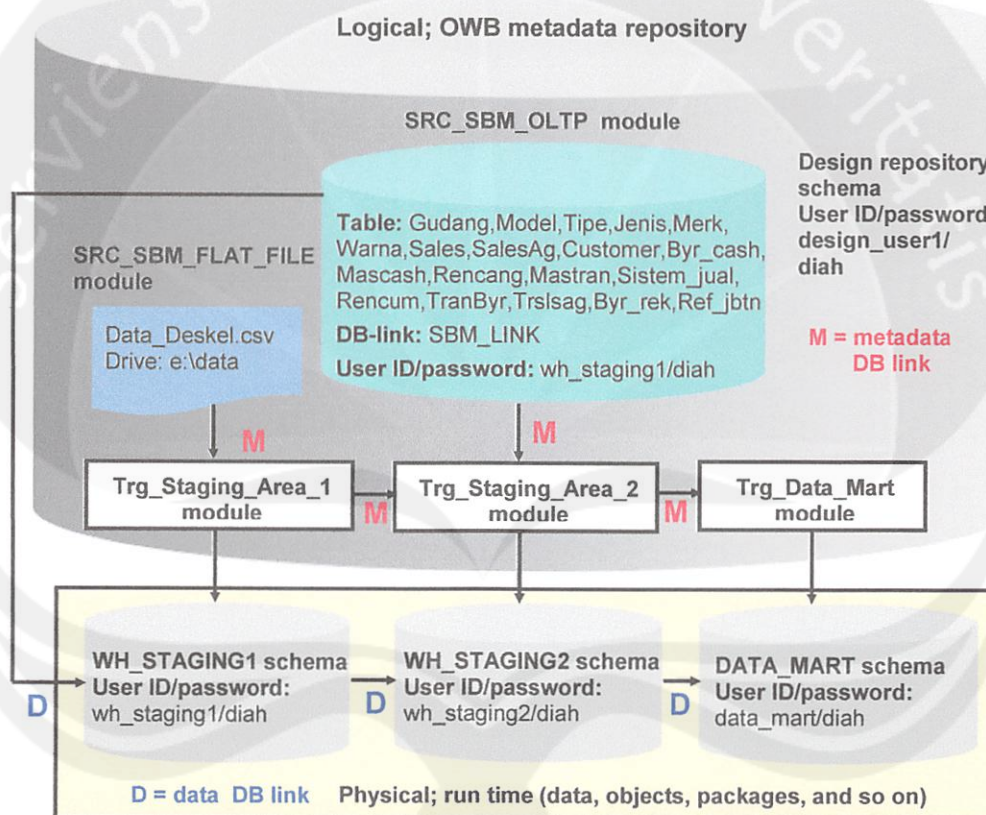
Gambar 4.4 Proses Pembangunan Data Warehouse

4.3.1 Loading Data Sumber ke Staging Area I

Dalam subbab ini akan dibahas tentang pembangunan data warehouse tahap I, yaitu proses pemindahan data dari sumber ke dalam tabel *staging* I. Dalam tahap ini yang dilakukan adalah proses transformasi data dari data sumber ke dalam database Oracle.

4.3.1.1 Menentukan Metadata Sumber

Metadata adalah salah satu bagian penting dalam perancangan *data warehouse*. Metadata dalam *data warehouse* serupa dengan kamus data atau katalog data pada DBMS. Metadata diperoleh dari berbagai sumber data yang berbeda. Gambar di bawah ini menunjukkan proses pendefinisian metadata:



Gambar 4.5 Pendefinisian Metadata Sumber

Dalam perancangan *data warehouse* ini, data-data sumber diambil dari 2 sumber data yang berbeda, yaitu:

4.3.1.1.1 Data Source Flat File

Flat file merupakan *file* yang tidak memiliki suatu format tertentu dan biasanya setiap *field*-nya hanya

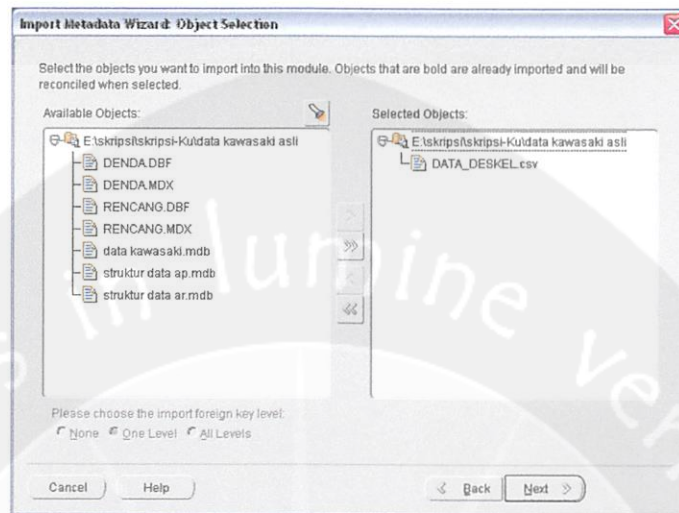
dibatasi oleh *delimiter*. *Delimiter* bisa berupa tanda petik ("), titik koma (;), koma (,), spasi, atau tab. Sedangkan tiap baris datanya dipisahkan oleh baris baru (<CR>). Dalam hal ini, *flat file* berisi data referensi tentang nama-nama kelurahan, kecamatan, dan kabupaten yang disimpan dengan nama DATA_DESKEL.csv.

Untuk mendefinisikan *metadata* sumber yang berupa *flat file* ke dalam *repository* digunakan *flat file module* yang akan menampung informasi-informasi tentang *flat file* yang akan di-*import*. Dalam penciptaan sebuah *flat file module* harus ditentukan alamat direktori dari file-file sumber dan nama lokasi *deployment* (*deployment location*). Sementara *deployment location* merepresentasikan lokasi fisik dalam OWB dimana *metadata* sumber akan di-*deploy*.

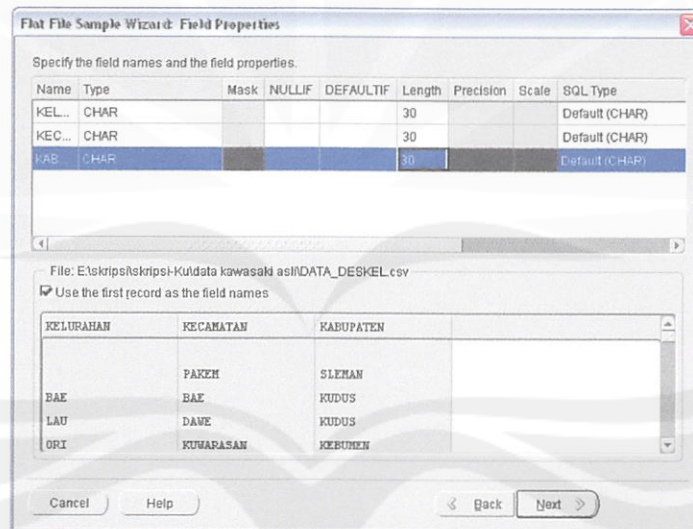
Flat file module yang digunakan diberi nama SRC_SBM_FLAT_FILE sedangkan *deployment location*-nya diberi nama SBM_FF_LOC. Dalam hal ini, SBM_FF_LOC adalah lokasi dimana *matadata* yang ada dalam modul SRC_SBM_FLAT_FILE akan di-*deploy* ke dalam *repository*.

Langkah selanjutnya adalah *import metadata* dari *flat file* ke dalam *flat file module* dan bila perlu melakukan *sampling* untuk mengatur konversi struktur data agar dapat dikenali oleh OWB.

Gambar berikut menunjukkan proses *import metadata* dan salah satu tahap dari proses *sampling*:



Gambar 4.6 *Import Metadata Flat File* ke dalam *Flat File Module*



Gambar 4.7 *Proses Sampling Flat File*

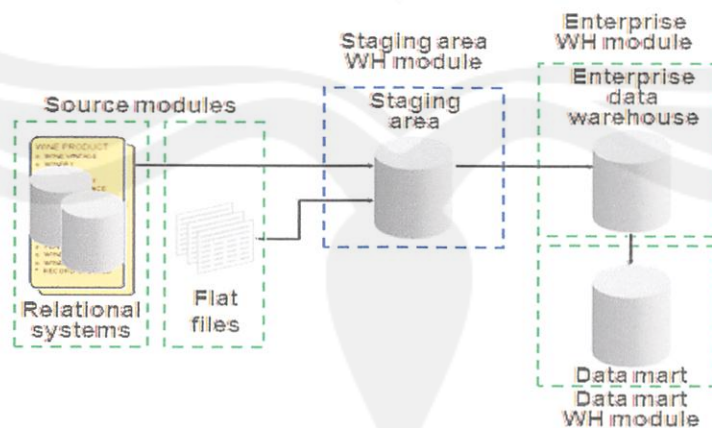
4.3.1.1.2 Sistem Transaksional

Sumber data ini berasal dari sistem transaksional perusahaan yang disimpan dalam bentuk *database* Microsoft Excel (*.mdb). Sumber data tersebut tersimpan dalam dua *database* yang berbeda, yaitu AP(Account

Payable).mdb dan *AR(Account Receivable)*.mdb. Database AP menyimpan data *supplier*, *stock* kendaraan, distribusi kendaraan, dan transaksi pembelian kendaraan bermotor dari *supplier*. Sedangkan database AR menyimpan data konsumen, gudang, *leasing*, sales, tipe kendaraan, transaksi pembayaran, dan penjualan baik tunai maupun kredit.

4.3.1.2 Memetakan Sumber ke dalam Staging Area

Pada sub bab berikut, akan dibahas mengenai proses penentuan area penyiapan data. Penentuan area penyiapan data yang dimaksud meliputi dua proses, yaitu mendefinisikan tabel *staging* dan *mapping* dari sumber ke dalam tabel *staging*. Tabel *staging* merupakan tabel hasil *copy* satu per satu dari tabel-tabel sumber. Tabel ini hanya berfungsi untuk pemindahan data. Tabel *staging* tidak mengandung *constraint foreign key* karena tabel ini hanya merupakan tabel sementara yang digunakan untuk memindahkan data. Gambar di bawah ini menunjukkan tujuan pendefinisian dari *staging* area.

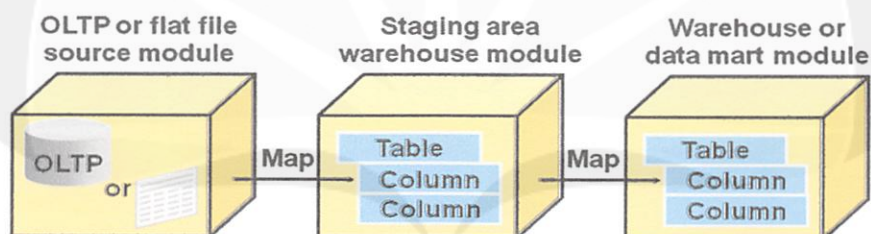


Gambar 4.8 Tujuan Pendefinisian *Staging* Area

Proses *mapping* secara formal menggambarkan operasi-operasi yang secara bersamaan menarik data dari sumber, mentransformasikan data yang dibutuhkan, dan *load* data ke dalam tabel yang menjadi target. Langkah-langkah dasar proses *mapping* dalam OWB meliputi:

- a. Pembentukan atau pemilihan *source module*.
- b. Pembentukan atau pemilihan *warehouse module*.
- c. Pembentukan atau *import* objek ke dalam *source module*.
- d. Pembentukan atau pemilihan *warehouse object*.
- e. Pemetaan *source object* ke dalam *warehouse object*.

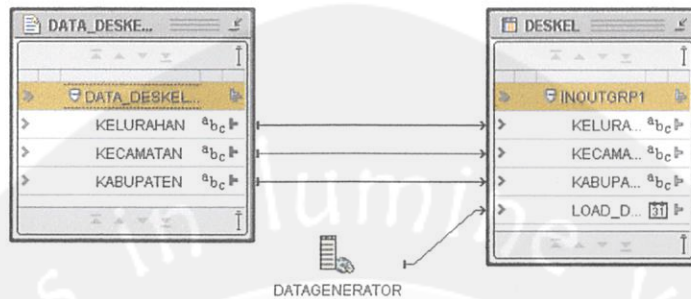
Mapping selalu dilakukan dalam *target module*. Gambar berikut menunjukkan proses *mapping* yang dilakukan dalam OWB:



Gambar 4.9 Proses *Mapping* dalam OWB

Pada kasus ini, proses *mapping flat file* yang diberi nama MAP_DESKEL dilakukan dalam *warehouse target* TRG_STAGING_AREA_1 yang sebelumnya telah dibentuk. Tabel hasil *mapping* dari file DATA_DESKEL tersebut akan dimodifikasi nama *field*-nya dan ditambah dengan *field* LOAD_DATE yang berisi tanggal saat *load* data dan tanggal diambil dari *sysdate*. Tabel tersebut diberi nama STG_DESKEL.

Berikut ini adalah gambar dari proses mapping *flat file* DATA_DESKEL:



Gambar 4.10 Proses Mapping Flat File DATA_DESKEL

Struktur data dari tabel hasil *mapping* dari sumber ke dalam *staging area* I adalah sebagai berikut:

Tabel 4.1 Struktur Data Tabel STG_DESKEL

NAMA FIELD	TIPE DATA
KELURAHAN	VARCHAR2 (30)
KECAMATAN	VARCHAR2 (30)
KABUPATEN	VARCHAR2 (30)
LOAD_DATE	DATE

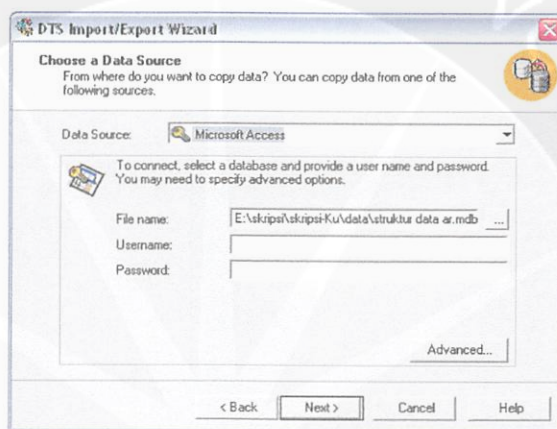
Lain halnya dengan *flat file*, proses *mapping* tabel-tabel yang ada pada sistem transaksional perusahaan akan disimpan ke dalam skema WH_STAGING1.

Proses *mapping* tersebut dilakukan dengan menggunakan *Data Transformation Server* (DTS) *Import/Export Wizard*. DTS *Import/Export Wizard* adalah suatu *tool* yang disediakan oleh SQL Server untuk mentransfer atau memindahkan data dari suatu *database* ke dalam *database* yang lain ataupun ke dalam format yang berbeda.

Berikut ini adalah langkah-langkah yang dilakukan dalam proses *mapping* dengan menggunakan DTS:

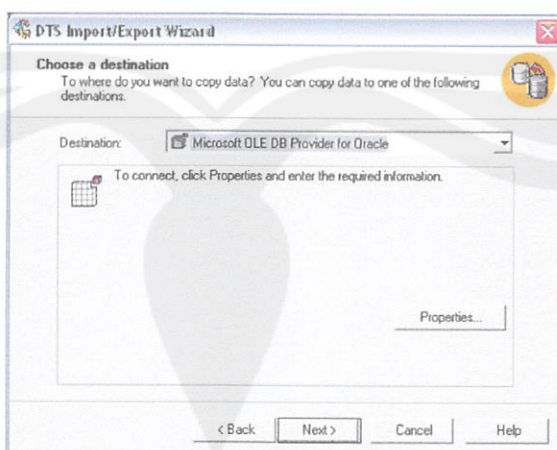
1. Pemilihan *Provider* untuk Sumber Data

Pemilihan *provider* harus sesuai dengan *database* sumber yang ada. Dalam kasus ini, karena sumber data berasal dari Microsoft Access maka pada bagian *Data Source* dipilihlah Microsoft Access. Kemudian pada bagian *file name* dipilihlah lokasi penyimpanan *database* AR.



Gambar 4.11 Halaman pemilihan *data source* pada DTS *Import/Export Wizard*

2. Pemilihan *Provider* untuk Tujuan Pemindahan Data



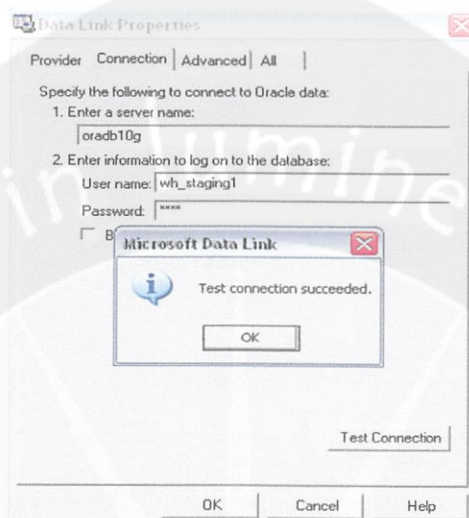
Gambar 4.12 Halaman pemilihan *provider* Oracle sebagai tujuan pemindahan data pada DTS *Import/Export Wizard*

3. Menentukan *Data Link Properties*

Nama server : oradb10g

User name : wh_staging1

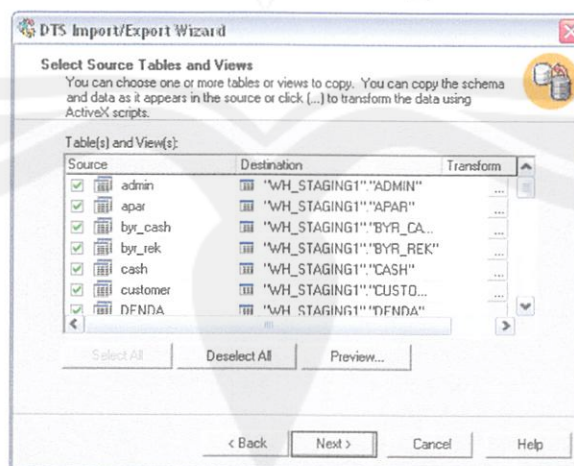
Password : diah



Gambar 4.13 Halaman pemilihan *provider* Oracle sebagai tujuan pemindahan data pada *DTS Import/Export Wizard*

4. Pemilihan Tabel-Tabel Sumber

Semua tabel yang ada pada *database* AR akan ditransfer ke dalam skema WH_STAGING1.



Gambar 4.14 Halaman pemilihan tabel-tabel sumber yang akan ditransfer ke dalam skema WH_STAGING1

Berikut ini adalah nama-nama tabel yang berasal dari sistem transaksional yang disimpan dalam skema WH_STAGING1:

Tabel 4.2 Daftar nama tabel pada sistem transaksional dan skema WH_STAGING1 Oracle

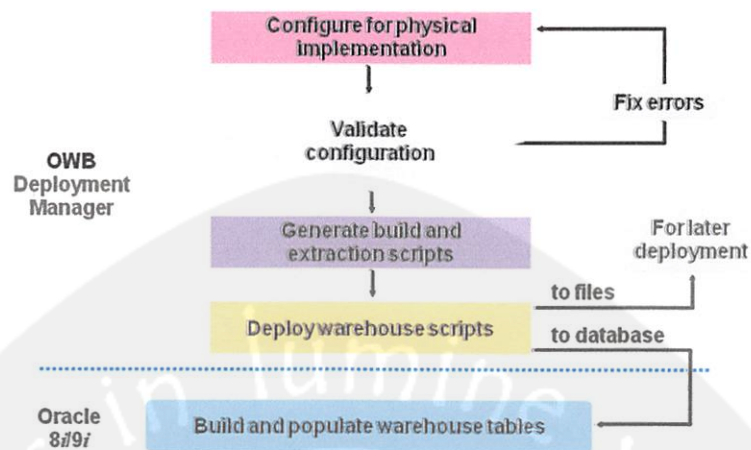
AP dan AR (ACCESS)	WH_STAGING1 (ORACLE)
admin	ADMIN
apar	APAR
APBELI	APBELI
apnonmob	APNONMOB
APTBSUPP	APTBSUPP
byr_cash	BYR_CASH
byr_rek	BYR_REK
cash	CASH
customer	CUSTOMER
DENDA	DENDA
DISTRIB	DISTRIB
gudang	GUDANG
jenis	JENIS
KOMIS	KOMISI
leasing	LEASING
mascash	MASCASH
mastran	MASTRAN
MBSTOCK1	MBSTOCK1
MBSTOCK2	MBSTOCK2
merk	MERK
model	MODEL
NAMATRAN	NAMATRAN
PEMBELI	PEMBELI
REF_JBTN	REF_JBTN
rekleas	REKLEAS
RENCANG	RENCANG

AP dan AR (ACCESS)	WH_STAGING1 (ORACLE)
RENCAP	RENCAP
rencum	RENCUM
sales	SALES
salesag	SALESAG
SISTEM_JUAL	SISTEM_JUAL
tipe	TIPE
TRANAP1	TRANAP1
TRANAP2	TRANAP2
tranbyr	TRANBYR
TRANLNS	TRANLNS
trslsag	TRSLSAG
uangmuka	UANGMUKA
warna	WARNA

4.3.1.3 Deploy Struktur Basis Data

Langkah selanjutnya setelah proses *mapping* selesai dilakukan adalah men-*deploy* struktur basis data. Proses *deploy* merupakan proses pembentukan skema target dari desain yang telah dibuat. Proses ini juga meliputi *generate script* yang akan dipakai untuk membentuk objek data seperti tabel, view, dan dimensi serta *generate script* untuk *load* data ke dalam objek data yang telah terbentuk. Seluruh informasi mengenai proses *deploy* secara detail akan disimpan di dalam *runtime repository*. Proses *deploy* dilakukan di OWB Deployment Manager.

Implementasi fisik dari proses *deploy* struktur basis data yang dilakukan di OWB dapat digambarkan seperti berikut:



Gambar 4.15 Implementasi Fisik dari Proses *Deploy* Struktur Basis Data

Langkah pertama yang dilakukan OWB adalah memeriksa apakah struktur basis data yang dibuat sudah tepat. Jika struktur basis data tidak valid, maka proses *deploy* gagal. Jika struktur basis data valid, maka OWB akan membuat *script* untuk ekstrak data. Setelah itu, OWB akan menciptakan *script* tersebut ke dalam *file* yang akan digunakan untuk proses *deployment* selanjutnya atau ke dalam basis data yang hasilnya akan disimpan dalam tabel *warehouse*.

Sebelum proses *deploy* dilakukan, seluruh lokasi tempat penyimpanan data harus di-*register* terlebih dahulu. Ketika *register* lokasi dilakukan, maka saat itu juga akan dilakukan proses spesifikasi informasi koneksi yang akan digunakan selama proses *deployment* untuk koneksi ke berbagai sumber data dan target.

Gambar berikut menunjukkan proses spesifikasi informasi koneksi:

Connection Information

Enter the warehouse runtime connection information

Connect As User:

Password:

Host Name:

Port Number:

Service Name:

Runtime Repository Owner:

Help OK Cancel

Gambar 4.16 Spesifikasi Informasi Koneksi

Bila koneksi berhasil, maka proses *deploy* struktur data tabel dapat dilakukan melalui OWB Deployment Manager dan secara otomatis *script* yang digunakan untuk proses *deploy* akan terbentuk.

Suksesnya proses *deploy* dalam staging area akan ditunjukkan seperti gambar di bawah ini:

Pre Deployment Generation Results for Runtime Repository access_user@localhost:1521:oradb10g

View: All Objects

Object	Deploy Action	Validation	Generation
MY_PROJECT			
LOC_TRG_STAGING_1			
TRG_STAGING_ARE...			
Table			
STG_DESKEL	Create	Success	Success
Drop			
Replace			
Upgrade			

Validation Script Impact Report

Object	Script Name	Object Type
STG_DESKEL	STG_DESKEL.ddl	TABLE
STG_DESKEL	STG_DESKEL_ANALYZE.ddl	ANALYZE

View Code Save As

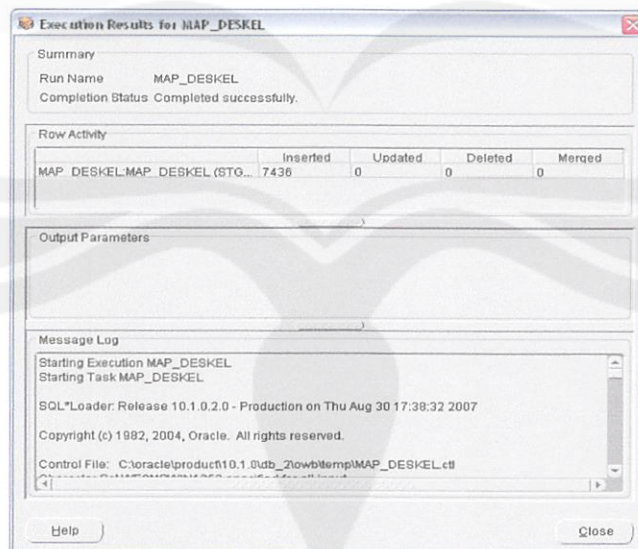
Help Deploy to file Deploy... Cancel

Gambar 4.17 Deploy Struktur Basis Data ke WH_STAGING1

Proses *deploy* di atas merupakan pengeksekusian *script* untuk membentuk tabel di *schema* tujuan berdasarkan *metadata* yang telah didefinisikan sebelumnya. *Script-script* tersebut akan di-*generate* secara otomatis oleh OWB apabila definisi *metadata* valid. Apabila *deploy* berhasil, maka tabel-tabel baru secara fisik akan tercipta di dalam *schema* WH_STAGING1.

4.3.1.4 Ekstrak Data Ke Tabel Staging

Pada tahap ini *script mapping* yang berisi definisi untuk *load data* telah terbentuk dalam OWB Repository, akan tetapi *script* tersebut masih belum dieksekusi ke dalam basis data target. Oleh sebab itu perlu dilakukan proses *deploy mapping flat file*. Setelah proses *deploy* berhasil, *script* tersebut dieksekusi untuk mengekstrak data dari *file* ke dalam tabel target yang telah terbentuk sebelumnya. Hasil eksekusi *script* untuk *mapping* MAP_DESKEL dapat dilihat pada gambar berikut:



Gambar 4.18 Hasil Eksekusi *Flat File Mapping*