BAB II
TINJAUAN PUSTAKA

2.1 Landasan Teori

Teori yang dibahas dalam penelitian ini antara lain tentang *War Room*, Integrasi Data, *Web Service, Data Warehouse*, *Business Dimensional Life Cycle Roadmap* , ETL (*Extract Transform Load*) dan *Tools* pendukung.

2.1.1 *War Room*

Menurut R.E Indrajit (2012) dalam bukunya Konsep Manajemen *Supply Chain*, *War Room* atau ruang perang pada dasarnya merupakan sebuah tempat dimana para manajemen inti dapat melakukan analisis secara kilat mengenai kondisi serius yang mendadak dihadapi perusahaan, agar dalam hitungan jam (bukan hari, minggu, atau bulan), manajemen perusahaan dapat mengambil keputusan secara strategis ataupun operasional terhadap langkah-langkah konkret yang harus dilakukan untuk mengurangi besarnya *damage* yang terjadi. Dalam penjelasannya desain awal *War Room* ini secara fisik dibagi menjadi dua bagian utama yaitu *Session Room* dan *Infrastructure Room*.

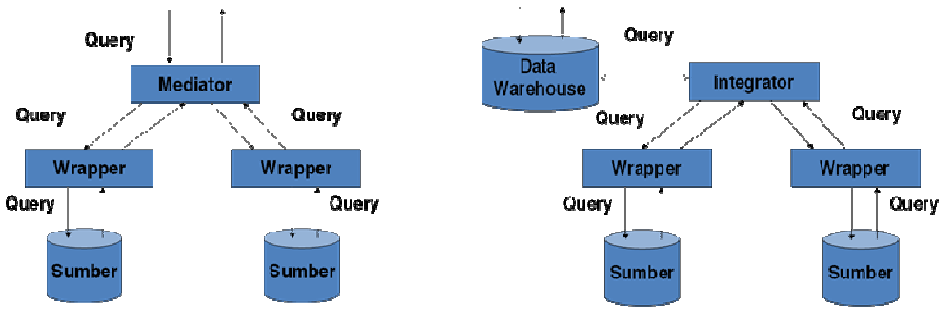
1) *Sessions Room*, merupakan ruang besar yang terbagi menjadi beberapa ukuran kecil, sedang, maupun besar tempat diskusi dilakukan. Di setiap ruangan disediakan berbagai peralatan presentasi multimedia yang dikontrol oleh fasilitas komputer (seperti *touch screen, digital board, desk control panel*, dan lain sebagainya). Melalui fasilitas ini, manajemen dapat mengakses data dan informasi apapun yang dibutuhkan secara *real-time,* dalam bentuk berbagai media seperti teks, grafik, gambar, audio, video, dan lain sebagainya.

2) *Infrastructure Room*, merupakan ruang yang sebenarnya berada di belakang layar, yang pada dasarnya di sinilah letak berbagai data internal maupun eksternal yang siap diolah menjadi informasi dan pengetahuan oleh berbagai jenis aplikasi-aplikasi penting (seperti *Executive Information System, Decision Support System)*. Tujuannya adalah agar semua data dan informasi yang dibutuhkan *Sessions Room* dapat disediakan oleh *infrastruktur Room*.

2.1.2 Integrasi Data

Menurut Lanzerini dalam bukunya *Data Integration A Theoritical Perspective Invite Talk PODS*, Integrasi data adalah proses menggabungkan atau menyatukan data yang berasal dari sumber yang berbeda dan mendukung pengguna untuk melihat kesatuan data. Integrasi data memusatkan pada perpindahan antar aplikasi dengan tujuan membagi data yang sama ke beberapa aplikasi yang berbeda.

Pada Gambar 2.1 beberapa pendekatan yang umum dilakukan untuk integrasi data, yang pertama dengan teknik mediasi / *virtual views* yaitu menggunakan *query* secara *direct access* ke *database*, yang kedua dengan teknik *warehousing* / *materialized views* yaitu menggunakan *query* yang hasilnya disimpan secara sentralisasi pada sebuah *data warehouse*.



Gambar 2.1 Integrasi Data (Lenzerini, 2002)

Dari sudut pandang teknis, integrasi data ini dipandang sebagai integrasi yang relatif lebih sederhana bagi para pengembang, mengakses basis data lebih mudah karena ada beberapa *tool* yang membantu memudahkan mempercepat *sharing* data, selain itu integrasi data ini tidak memerlukan perubahan aplikasi.

Integrasi data dibutuhkan seiring dengan perkembangan organisasi dan meningkatnya bisnis proses pada institusi tersebut yang saling membutuhkan data dan informasi dari divisi atau unit-unit yang berada pada organisasi tersebut.

2.1.3 *Web Service*

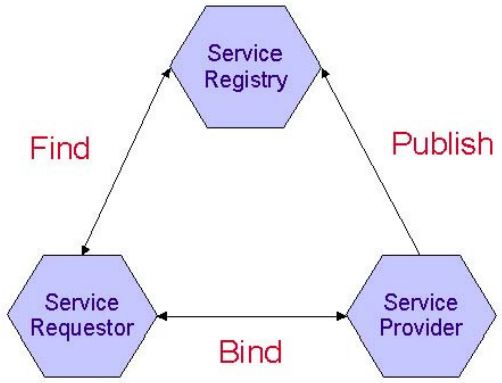
Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interoperabilitas* dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh *web site* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML (*Extensible Markup Language*), sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*. *Web service* bertujuan untuk meningkatkan kolaborasi antar pemrogram, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa menggunakan *web service* sebagai penyedia layanan dalam bentuk informasi kepada sistem yan lain, adalah sebagai berikut:

1) *Web service* dapat mentransformasikan beberapa bisnis logik dan objek yang terpisah dalam satu ruang lingkup dengan tingkat keamanan yang baik.

2) *Web service* memiliki kemudahan dalam proses *deployment*, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di *upload* ke *web server* dan siap diakses oleh pihak yang diberikan otorisasi.
3) *Web service* berjalan di port-80 yang merupakan protokol standar HTTP (*HiperText Transfer Protocol*), dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

Web service memiliki tiga entitas dalam arsitekturnya, yaitu *Service Requester* (peminta layanan), *Service Provider* (penyedia layanan), *Service Registry* (daftar layanan), seperti gambar 2.2.



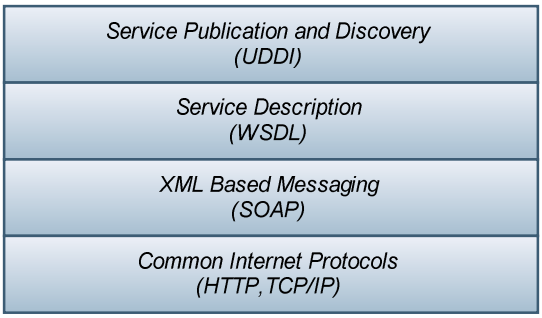
Gambar 2.2 Arsitektur *Web Service* (Gardner,2001)

1) *Service Requestor* : Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.
2) *Service Provider* : Berfungsi untuk menyediakan layanan dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
3) *Service Registry* : Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan yang telah diregister.

Sedangkan operasi dasar untuk Interaksi antar role, yaitu :

1) *Publish :* Menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2) *Find : Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3) *Bind : Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan *binding* ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider.*

Gambar 2.3 merupakan komponen *Web service* yang menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur *web service* (Tidwell,2001).

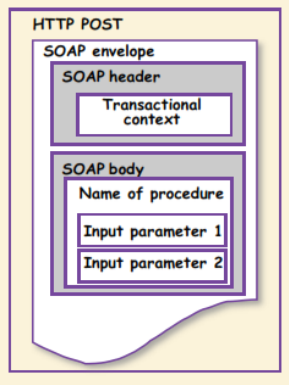


Gambar 2.3 Komponen *Web Service* (Tidwell,2001)

1) Layer 1 : Protokol internet standar ang digunakan sebagai sarana transportasi adalah HTTP, TCP/IP.
2) Layer 2 : *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML (*Extensible Markup Language*) untuk proses pertukaran data/informasi antar layanan.
3) Layer 3 : *Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML (*Extensible Markup Language*) yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.

4) Layer 4 : *Universal Description Discovery and Integration* (UDDI), merupakan direktori pusat untuk diskripsi layanan.

Gambar 2.4 menjelaskan format pertukaran informasi/komunikasi antar sistem melalui HTTP menggunakan struktur XML untuk proses transaksi antar sistem baik dalam proses *request* maupun *response* dalam *web service*.



Gambar 2.4 Struktur XML SOAP (Papazoglou,2008)

2.1.4 *Data Warehouse*

Data warehouse merupakan pondasi dari semua proses EIS (*Executive Information System*) karena mempunyai satu sumber data yang terintegrasi dengan tingkat granularitas yang tepat, beberapa definisi dari *data warehouse* antara lain :

Menurut Connolly & Begg (2005), *data warehouse* adalah *subject-oriented*, *integrated*, *time variant* dan *non volatile* dalam pengumpulan data untuk mendukung pembuatan pengambilan keputusan.

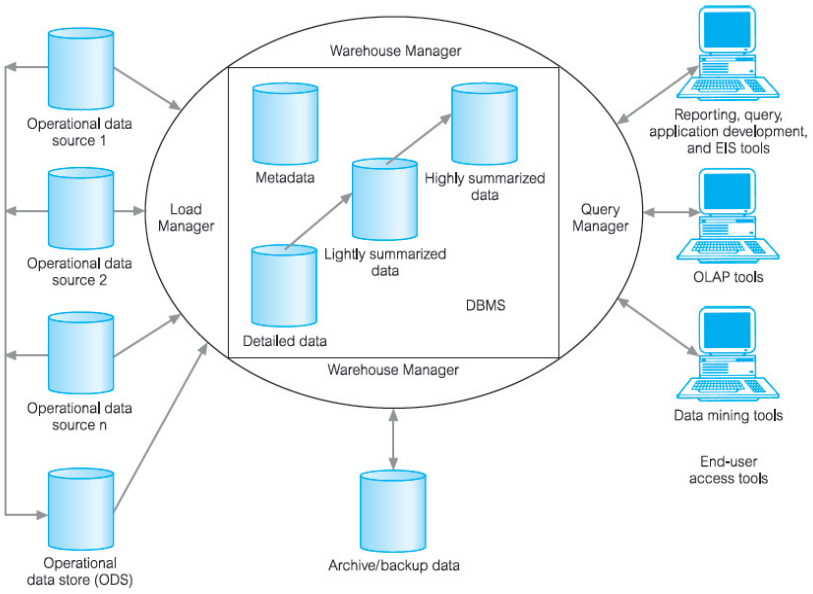
Menurut W.H. Inmon (2005), *data warehouse* adalah koleksi data yang mempunyai sifat berorientasi subjek, terintegrasi, *time variant* dan bersifat tetap dari koleksi data dalam mendukung proses pengambilan keputusan manajemen.

Dari definisi-definisi yang dijelaskan tadi, dapat disimpulkan *data warehouse* adalah database yang saling bereaksi yang dapat digunakan untuk *query* dan analisis yang bersifat *Subject oriented*, *Integrated*, *Time-variant* dan *Non Volatile. Data*

warehouse merupakan suatu paradigma baru pengambilan keputusan dimana penggunanya dapat menemukan informasi yang terpisah dengan database operasional.

1) Komponen *Data Warehouse*

Komponen-komponen yang terdapat di dalam arsitektur *data warehouse* menurut Connolly & Begg dapat dijelaskan sesuai dengan gambar 2.5 berikut :



Gambar 2.5 Arsitektur *Data Warehouse* (Connolly & Begg, 2010)

(1) Operational Data

Sumber-sumber data yang ada di *data warehouse* berasal dari data operasional pada *database* hirarki dan *database* jaringan, data departemental, data pribadi (disimpan di dalam *workstation* dan *private serve),* Sistem eksternal seperti *internet*, *database* komersial, atau *database* yang berhubungan dengan organisasi dari *supplier* atau konsumen.

(2) *Operational Data Store*

Sebuah *Operational Data Store* (ODS) adalah sebuah penyimpanan sementara dari data operasional. ODS biasanya melakukan penstrukturan dan penyediaan data seperti halnya sebuah *data warehouse*, tetapi sebenarnya bertindak secara sederhana sebagai suatu tempat penampungan sementara sebelum data akan dipindahkan ke dalam *data warehouse,* namun bersifat *volatile* yang masih bisa diupdate sebelum akhirnya disimpan kedalam *data warehouse*.

(3) *Load Manager*

Load manager atau biasa disebut komponen *fronted,* melakukan sebuah operasi terkait dengan ekstraksi dan pemuatan data ke dalam *warehouse*. Data mungkin diekstrak secara langsung dari sumber data atau dari *Operational Data Store*.

(4) *Warehouse Manager*

Warehouse manager melakukan semua operasi yang berhubungan dengan pengelolaan data di dalam *warehouse*. Komponen ini dikonstruksikan dengan menggunakan *vendor data management* dan *custom built program.*

(5) *Query Manager*

Query Manager yang juga disebut komponen *back end*, melakukan semua operasi yang berhubungan dengan pengelolaan *user queries.* Komponen ini dibangun dengan menggunakan *vendor user-end data access tools*, *data warehouse monitoring*, fasilitas *database*, dan *custom built program.*

(6) *Detailed Data*

Area dari *data warehouse* ini menyimpan semua detail data di dalam skema *database*. Dari beberapa kasus yang ada, detail data tidak di simpan secara *online*, tetapi dibuat melalui agregasi data pada tingkatan-tingkatan detail berikutnya.

(7) *Lightly* dan *Highly Summarized Data*

Area ini menyimpan semua *lightly* dan *highly summarized* data yang dihasilkan oleh *warehouse manager*. Area dari *data warehouse* ini adalah sebuah tempat untuk menampung sementara sebelum dilakukannya perubahan secara berkelanjutan untuk merespon perubahan profil *query*.

(8) *Archive / Backup Data*

Area dari *data warehouse* ini menyimpan semua detail dan ringkasan data yang bertujuan untuk melakukan *archiving* dan *backup.*

(9) *Metadata*

Area dari *warehouse* ini menyimpan sebuah definisi *metadata* (data dari data), yang digunakan oleh semua proses didalam *warehouse*.

(10) *End-User Access Tools*

Tujuan dari *data warehousing* adalah untuk menghasilkan sebuah informasi untuk *end-user* dalam strategi pembuatan keputusan menggunakan *end-user access tools*. Ada lima kategori utama dari *end-user access tools:*

a. Reporting and query tools

Reporting tools digunakan untuk menghasilkan laporan operasional regular atau mendukung *high-volume batch job*. *Query tools* dirancang untuk menerima SQL dalam proses *query* data didalam *data warehouse.*

b. Application development tools

Aplikasi yang sesuai dengan kebutuhan *user*, dirancang untuk *client server.*

c. *Executive information system (EIS) tools*.

EIS digunakan untuk memberikan dukungan kepada manajemen *high-level* dalam menentukan atau pembuatan keputusan yang stategis yang dapat diteruskan kepada semua tingkat manajemen.

d. *Online analytical processing (OLAP) tools*

OLAP berbasis pada konsep *database multidimensi* yang dapat digunakan untuk menganalisis data dengan menggunakan sebuah *view* terhadap data yang kompleks dan *multidimensional*.

e. Data mining tools

Data mining tools digunakan untuk melakukan pengumpulan data historis dan menemukan keteraturan atau pola dan korelasi terhadap sejumlah kejadian, yang hasilnya dapat digunakan sebagai informasi dan *knowledge* yang berguna.

2) Karakteristik *Data Warehouse*

Menurut Inmon *Data Warehouse* mempunyai karakteristik :

(1) *Subject Oriented*

Data Warehouse berorientasi subyek artinya *data warehouse* didesain untuk menganalisis data berdasarkan subyek-subyek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu.

(2) *Integrated*

Data Warehouse dapat menyimpan data-data yang berasal dari sumber-sumber yang terpisah kedalam suatu format yang konsisten dan saling terintegrasi.

(3) *Time-variant*

Seluruh data pada *data warehouse* dikatakan akurat pada rentang waktu tertentu.

(4) *Non Volatile*

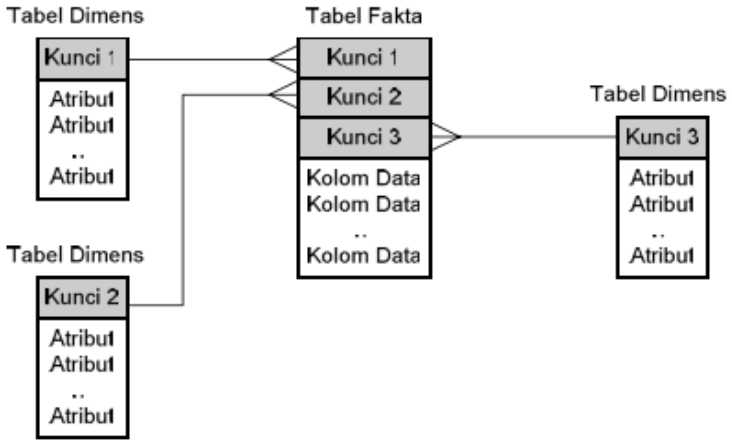
Data Warehouse tidak diupdate secara *real time* tetapi di refresh dari sistem operasional secara reguler. Data yang baru selalu ditambahkan sebagai suplemen bagi *database* itu sendiri dari pada sebagai sebuah perubahan.

3) Model Dimensional *Data Warehouse*

Ada beberapa teknik pendekatan yang dipergunakan untuk merancang model dimensional pada *data warehouse* yaitu :

(1) *Star Schema*

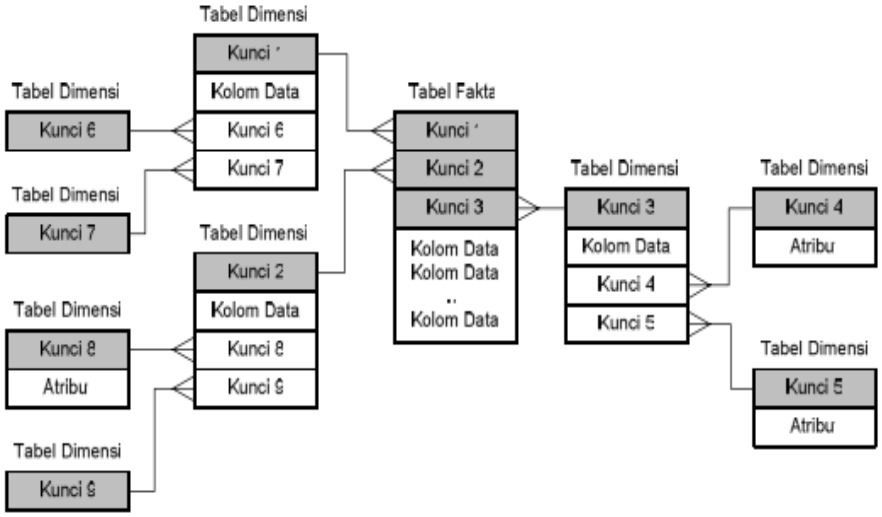
Skema ini mengikuti bentuk bintang, terdapat satu tabel fakta di pusat bintang dengan beberapa tabel dimensi yang mengelilinginya. Tabel fakta memiliki beberapa *key* yang merupakan kunci indek individual dalam tabel dimensi, seperti pada gambar 2.6.



Gambar 2.6 Model *Star Schema* (Connolly & Begg, 2010)

(2) *Snowflake Schema*

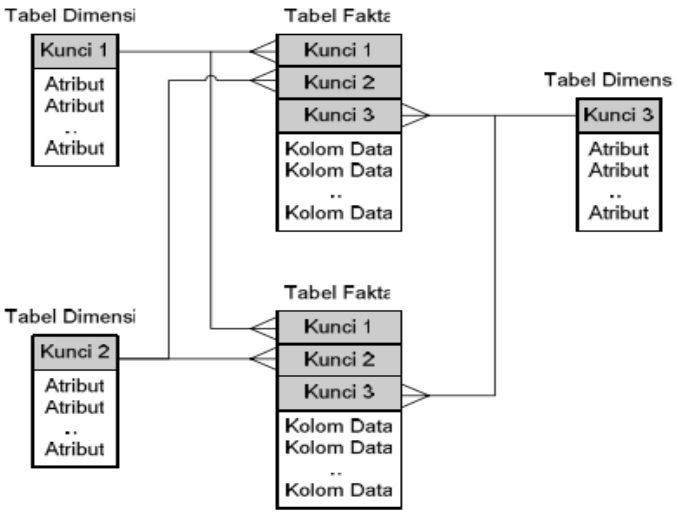
Snowflake Schema merupakan perluasan dari model dimensional *data warehouse star schema* dengan tambahan beberapa tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta. Namun tabel dimensi tersebut berhubungan dengan tabel-tabel dimensi yang lain, ilustrasi skema seperti pada gambar 2.7.



Gambar 2.7 Model *Schema Snowflake* (Connolly & Begg, 2010)

(3) *Fact Constellations*

Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi secara bersama-sama, seperti pada gambar 2.8.

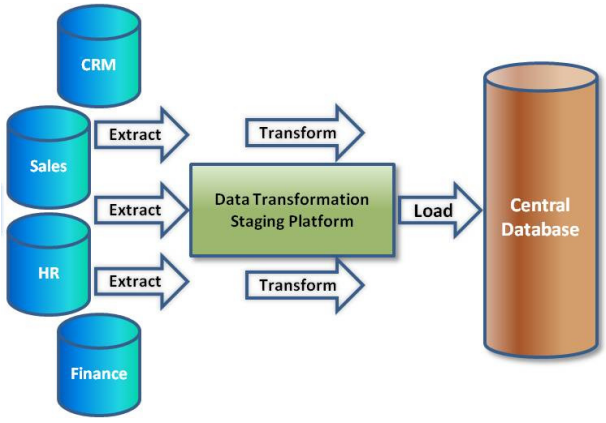


Gambar 2.8 Model *Fact Constellation* (Connolly & Begg, 2010)

2.1.5 ETL (*Extract Transform Load*)

Menurut Ponniah (2001), ETL (*Extract Transform Load*) merupakan proses yang sangat penting dalam pembangunan sebuah *data warehouse*, dengan ETL inilah data dari operasional dapat dimasukkan ke dalam *data warehouse*. ETL juga dapat digunakan untuk mengintegrasikan data dengan sistem yang sudah ada baik data internal maupun eksternal. Tujuan ETL adalah untuk mengumpulkan, menyaring, mengolah, dan menggabungkan data-data yang relevan dari berbagai sumber untuk disimpan ke dalam *data warehouse*.

Hasil dari proses ETL adalah dihasilkannya suatu data yang memenuhi kriteria *data warehouse* yang sudah ditetapkan pada proses transformasi seperti penamaan field, panjang karakter, tipe data. Yang terpenting adalah hasilnya merupakan data yang konsisten, terpadu, terangkum dan memiliki struktur yang digunakan untuk keperluan proses-proses analisis selanjutnya. Penggambaran proses mulai dari *Extraction, Transformation* sampai dengan *Load* dari beberapa sumber data yang terpisah, baik data operasional maupun data eksternal lainnya ke dalam *data warehouse* dapat dilihat pada gambar 2.9.



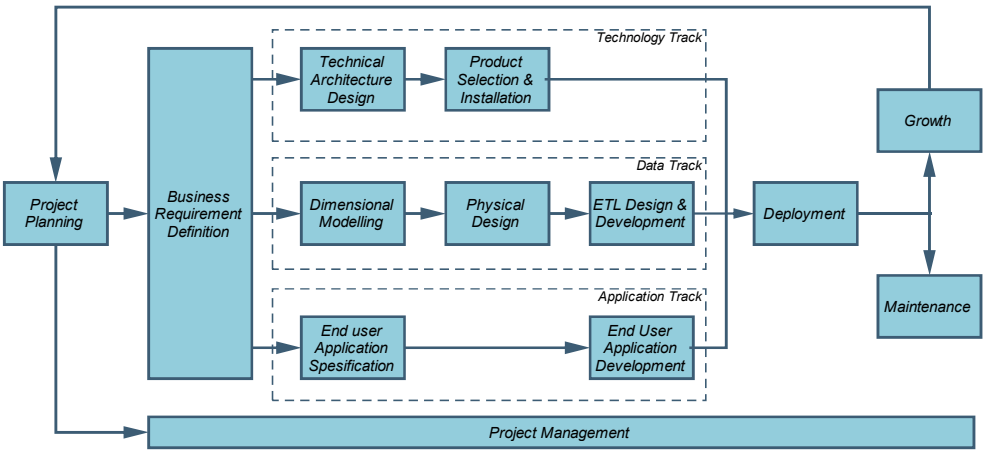
Gambar 2.9 Proses ETL (IMC's ETL Process Management services,2011)

Secara detail proses ETL yang umum digunakan dalam pembuatan *data warehouse* terdiri dari proses *Extraction, Transformation,* dan *Load* sebagai berikut :

1) *Extraction*, proses ini membaca struktur data dan isinya dari sumber data yang selanjutnya akan didefinisikan ke dalam sebuah Skema Standar ETL *Tool*.
2) *Transformation*, proses ini untuk menyesuaikan apa saja yang akan di *load* ke dalam target, pada proses transform bisa berupa *simple copy, join* dari sumber data lain, agregasi, *sorting*, *filter*, dan lain-lain.
3) *Load*, merupakan proses terakhir yang melakukan penyimpanan data ke dalam target *data warehouse.*

2.1.6 *Business Dimensional Life Cycle Roadmap*

Menurut Kimball & Ross (2010), pendekatan *Business Dimensional Life Cycle* merupakan pendekatan dalam membangun *data warehouse*. Diagram yang menjelaskan suatu urutan tugas yang dibutuhkan mulai dari *planning*, *requirement*, desain, pengembangan, dan implementasi *data warehouse* yang efektif, seperti gambar 2.10 berikut :



Gambar 2.10 *The Kimball Lifecycle Diagram* (Kimball & Ross, 2010)

Kimball & Ross membagi urutan tugas *Business Dimensional Life Cycle* tersebut menjadi 6 langkah, yakni:

1) *Project Planning*

Roadmap pertama yang berfokus untuk mendapatkan program/proyek yang diluncurkan, termasuk *Asessing Readines* (kesiapan organisasi), *scoping* (ruang lingkup), *justification* (dukungan manfaat dan biaya), *staffing* (dukungan integrasi tim) dan *Developing and Maintaining the Project Plan* (pengembangan keterlibatan semua tugas).

2) *Business Requirements Definition*

Memunculkan suatu kebutuhan bisnis adalah tugas kunci dalam *lifecycle* Kimball, persyaratan dikumpulkan untuk menentukan faktor-faktor kunci yang berdampak bisnis dengan berfokus pada apa yang pengguna bisnis lakukan. Peluang utama di seluruh perusahaan diidentifikasi, diprioritaskan berdasarkan nilai bisnis dan kelayakan, dan kemudian persyaratan yang rinci berkumpul untuk iterasi pertama dari *data warehouse*.

3) *Data Track*

Terdiri dari tahap *Dimensional Modelling, Physical Design* dan *ETL Design & Development. Data track* dimulai dengan desain model target dimensi untuk menangani kebutuhan bisnis, dengan tetap memperhatikan realitas data yang mendasarinya. Model dimensi dikonversi menjadi desain fisik di mana kinerja strategi tuning dipertimbangkan, kemudian dilakukan *extract, transform,* dan *load* (ETL) sistem desain dan tantangan *development* yang ditangani.

4) *Application Track*

Terdiri dari tahap *End user Application Spesification* dan *End User Application Development. Application Track* mengidentifikasi dan membangun berbagai

aplikasi *end user*. Aplikasi *end user* yang dibuat berupa aplikasi untuk keperluan lain yang memanfaatkan dibangunnya *data warehouse*.

5) *Technology Track*

Terdiri dari tahap *Technical Architecture Design* dan *Product Selection & Installation*. Lingkungan *data warehouse* mewajibkan integrasi dari berbagai teknologi, *data stores*, dan *metadata* yang terkait. Trek teknologi dimulai dengan desain sistem arsitektur untuk membuat *shopping list* dari kemampuan yang dibutuhkan, dilanjutkan dengan pemilihan dan pemasangan produk yang memenuhi kebutuhan-kebutuhan arsitektur.

6) *Deployment, Maintetance, and Growth*

Fase *Deployment* perancangan teknologi, data dan aplikasi yang sudah dilakukan pada tahap sebelumnya. Pada *fase maintenance and growth* tim proyek memfokuskan pada persyaratan yang akan dihadapi, penyampaian yang signifikan dan / atau risiko dalam usaha penerapan. Namun dalam pembahasan pada penelitian ini dibatasi sampai dengan perancangan sehingga pembahasan tidak menyangkut tahap *deployment, maintenance and growth*.

2.1.7 Pentaho

Pentaho adalah kumpulan aplikasi yang berjalan di atas *platform* Java. Aplikasi-aplikasi Pentaho dikembangkan oleh Pentaho Corporation yang berpusat di Orlanda, Amerika Serikat. Versi Pentaho ada yang dalam bentuk *community edition* yang bersifat *free open source software* (FOSS), ada juga yang dalam bentuk *Service Level Agreement* (SLA) dan dipaketkan dalam versi *Enterprise Edition* yang sifatnya *annual subscription* atau perlu kontrak tahunan. Menurut Maria Carina Roldan

(2010) dalam bukunya *Pentaho 3.2 Data Integration*, beberapa macam aplikasi yang didukung oleh Pentaho antara lain :

1) Pentaho *Data Integration* (PDI) / Kettle
2) Pentaho *Reporting*
3) Pentaho *Analysis* / Mondrian OLAP *Server*
4) Pentaho *Data Mining* / Weka
5) Pentaho *BI Server / Platform*

Yang dibahas dalam penelitian ini berfokus pada Pentaho *Data Integration* (PDI) atau disebut dengan *Kettle* dengan versi *Community edition*, yaitu :

1) Utilitas ETL (*Extract, Transform and Load*) bersifat *open source*.
2) *Designer* GUI yang intuitif dan sangat mudah digunakan.
3) *Multi Platform.*
4) *Script* ETL dapat disimpan dalam bentuk *filesystem* maupun *repository.*
5) Mendukung *multi pipelining* sehingga *load balance* maupun optimasi pekerjaan *data warehouse* dapat dilakukan dengan mudah.
6) Mendukung *clustering (master-slave) engine* ETL.
7) Terdiri atas lebih dari 200 steps yang mencakup *job (workflow controll)* dan *transformation (data worfklow).*
8) Mendukung *Apache Virtual Filesystem* (Apache VFS) sehingga *filesystem* seperti HTTP Webdav, FTP, SFTP, dan lain sebagainya dapat dengan mudah diakses dengan konfigurasi yang minimal.

Sehingga untuk keperluan proses integrasi data, *Pentaho Data Integration* sangat tepat digunakan untuk melakukan proses Ekstrak data yang berasal dari beberapa

sumber sekaligus, karena mempunyai fasilitas *multi pipelining* dimana instruksi dapat dilakukan dalam satu waktu yang sama.

2.2 Penelitian Terkait

Berikut adalah beberapa penelitian sebelumnya yang terkait dengan topik permasalahan tentang pengintegrasian data sistem informasi, antara lain implementasi *web service* dan *data warehouse* :

1) Implementasi Integrasi Data Antar Sistem Informasi Untuk Mendukung *Decission Support System*, (Mugi Sugiarto, Pelita Fajarhati,2008).

   Penelitian ini menyampaikan suatu konsep menggabungkan data-data heterogen yang berasal dari berbagai sumber aplikasi sistem informasi, yang tidak hanya dengan melakukan perintah *query* secara *direct access* tapi sudah mempunyai gagasan dengan melakukan proses ETL kedalam *data warehouse* dan pemanfaatan teknologi web, untuk mengurangi redundansi dan duplikasi data.

   Namun dalam uraiannya hanya berupa konsep yang tidak menjelaskan bagaimana arsitektur dan langkah untuk melakukan perancangan integrasi data pada *data warehouse* dan teknik pemanfaatan data untuk distribusi datanya.

2) Analisa dan Perancangan *Web Service* untuk Sistem Informasi Universitas (Budi Santosa, 2008)

   Dalam penelitian ini menerapkan konsep XML *Web Service* sebagai solusi dalam pengembangan sistem informasi universitas, fokus utama penelitian adalah pada eksplorasi kemampuan XML *Web Services* terutama dalam hal integrasi data dan fungsi dalam lingkup universitas, kolaborasi antar pemrogram dan kemampuannya untuk diakses oleh bermacam-macam aplikasi dan *device*.

Hal-hal yang belum dieksplore dalam penelitian ini adalah bagaimana data dari operasional yang dihimpun sudah dilakukan *Ektract Transform dan Load* (ETL) menjadi data yang siap digunakan sebagai sumber distribusi data untuk kepentingan *end user* lainnya.

3) Perancangan *Data warehouse* dengan Pendekatan *Enterprise Architecture* (Kusuma Ayu LaksitoWening, 2010)

   Dalam penelitian ini menjelaskan perancangan *data warehouse* menggunakan model rancangan *enterprise architecture*, yang digunakan dalam penelitian ini sepenuhnya mengadopsi pada penerapan *Framework* Zachman sebagai penggambaran proses bisnis yang ada di dalam perusahaan.

   Saran dalam penelitian ini masih berupa pemodelan awal untuk perancangan arsitektur *enterprise* yang utuh dan lebih lengkap, sehingga untuk jangka panjang perlu dikembangkan lagi kedalam bentuk *blue print* yang lebih detil serta berisikan daftar rencana dalam pengembangan arsitektur *enterprise* yang nantinya dapat diterapkan oleh perusahaan-perusahaan khususnya di Indonesia.

Sesuai dengan tujuan penelitian dan pembahasan kajian pustaka yang telah dilakukan, peneliti akan melakukan perencanaan dan analisis kebutuhan informasi untuk merancang sistem pengintegrasian data pada media penyimpanan *data warehouse*. Hasil yang diharapkan adalah suatu rancangan arsitektur integrasi data pada sebuah *data warehouse* yang dioptimasi untuk keperluan analisis data perusahaan yang diolah pada unit *War Room* Telkom dan distribusi data lainnya dengan teknologi *web service*.

Biasanya perusahaan menyimpan data dan informasi ke dalam dua bentuk, yang pertama adalah dalam bentuk database operasional dimana data yang disimpan

dapat diedit atau dihapus. Sedangkan pada *data warehouse* data didapatkan dari sumber database operasional yang telah diproses, dimana data yang sudah disimpan bersifat *Non volatile* yang tidak diupdate, namun direfresh dari sistem operasional untuk penambahan data yang baru. Hal-hal yang mendasari penggunaan *data warehouse* adalah :

1) Membuat informasi pada perusahaan atau organisasi mudah terakses (*accessible*). Isi data pada *data warehouse* dapat mudah dipahami, dinavigasi, dan akses data yang cepat.
2) Membuat informasi pada perusahaan atau organisasi konsisten. Informasi dari salah satu bagian (divisi) perusahaan atau organsisasi dapat dicocokkan dengan informasi dari bagian (divisi) lain. Jika pada divisi dalam suatu perusahaan atau organisasi memiliki nama yang sama, maka mereka harus menyamakan kedua informasi tersebut. Sebaliknya, jika pada divisi dalam suatu perusahaan atau organisasi memiliki nama data yang berbeda, maka mereka harus membedakan kedua informasi tersebut. Informasi yang konsisten berarti informasi berkualitas tinggi. Ini berarti bahwa semua informasi tersebut dicatat dan selesai.
3) Sumber informasi memiliki ketahanan dan adaptif, *data warehouse* dirancang untuk perubahan yang bersifat terus menerus, ketika pertanyaan-pertanyaan baru atau kebutuhan baru dibutuhkan dari *data warehouse*, maka data yang ada dan teknologi tidak akan terganggu maupun berubah.
4) Melindungi aset perusahaan atau organisasi, *data warehouse* tidak hanya mengontrol akses ke data secara efektif, tetapi memberikan visibilitas yang besar ke dalam penggunaan dan penyalahgunaan data.
5) Sebagai dasar dalam pengambilan keputusan, *data warehouse* memiliki data yang benar di dalamnya untuk mendukung pengambilan keputusan. keputusan

yang dibuat setelah *data warehouse* telah mengajukan buktinya atau menampilkan data secara rinci.

Untuk menghasilkan cara yang lebih baik dalam mengelola data, yang dilandasi oleh beberapa penelitian terkait yang sebelumnya dilakukan oleh peneliti lain untuk membangun suatu integrasi data yang berasal dari berbagai sumber data operasional perusahaan. Ada yang mendapatkan data dengan *query* secara *direct access* ke sumber data, selanjutnya ada pula yang menyimpannya ke dalam *data warehouse* namun datanya hanya untuk kepentingan sendiri dan data yang sudah tersimpan tidak dimanfaatkan untuk aplikasi atau sistem informasi lainnya, sehingga memungkinkan unit-unit lain juga membangun *data warehouse* sendiri. Dalam penelitian lain ada yang menggunakan cara untuk mendapatkan data melalui aplikasi *web service* dalam melakukan *query* ke sumber data operasional, maksudnya untuk menghindari kendala perbedaan *platform*, namun setelah dianalisis lebih lanjut ternyata cara ini bila digunakan untuk melayani banyak *requester* cukup dapat mengganggu performa dari *server* sumber datanya.

Terkait dengan kebutuhan terhadap data yang terintegrasi yang dapat digunakan menjadi multifungsi selain untuk keperluan Sitem Informasi Eksekutif dan kebutuhan penggunaan oleh sitem informasi lainnya, maka sudah saatnya kita menganalisa dang menggabungkan antara *resource* yang dimiliki saat ini dengan konsep teknologi yang dapat menunjang perancangan, yaitu menggunakan model *data warehouse* dan model aplikasi *web service*.

Berdasarkan kondisi saat ini, data operasional yang tersimpan dalam masing-masing *database* eksisting dan ketersediaan data *legacy* EIS, selanjutnya dalam penelitian ini data-data tersebut akan dikelola kedalam *data warehouse* dengan

model interaksi layanan dengan *web service*. Hasil perancangan yang dimaksud adalah berupa sebuah *data warehouse* yang strukturnya dikembangkan secara spesifik untuk keperluan analisis pihak eksekutif, sedangkan untuk keperluan distribusi data yang sudah terkumpul di *data warehouse* untuk unit lain yang membutuhkan data akan diproses melalui *web service*. Akhirnya dari penggabungan tersebut diharapkan *War Room* Telkom akan mempunyai suatu rancangan arsitektur integrasi data berbasis *data warehouse* dan *web service* yang dapat melakukan pengumpulan data dari berbagai sumber juga untuk keperluan melayani distribusi data pengguna lainnya.