

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian atau perancangan yang berkaitan dengan pangkalan data (pangkalan data) sudah beberapa kali dilakukan, seperti penelitian yang berjudul “Perancangan pangkalan data Sistem Informasi Eksekutif dengan Metode Kimball (Kasus pada Universitas Muhammadiyah Magelang)” telah dilakukan Setiawan, Noor Akhmad, dan Sri Suning Kusumawardani (2014). Perancangan ini menggunakan metode 9 Langkah (Nine Step Methodology) dari Kimball, Tujuan menggunakan metode ini adalah karena memiliki tahapan perancangan yang dimulai dari kebutuhan, sehingga sistem mampu menyajikan informasi dengan struktur lengkap dan sesuai dengan kebutuhan pengambil keputusan. Hasil perancangan berupa pangkalan data yang telah digunakan untuk membangun salah satu bagian Sistem Informasi Eksekutif yaitu bidang Akademik (SIE-Akademik), Sistem ini mampu menyajikan data yang terintegrasi dan sapat ditampilkan sesuai dimensi faktual yang diinginkan berupa ringkasan IP mahasiswa selama 7 tahun.

Khusnul Khotimah dan Sriyanto (2017), melakukan perancangan yang berjudul “PERANCANGAN DAN IMPLEMENTASI DATA WAREHOUSE UNTUK Mendukung Sistem Akademik (STUDI KASUS PADA STKIP MUHAMMADIYAH KOTABUMI)”. Perancangan ini menggunakan metode Kimball dengan metodologi “nine-step methodology”, untuk perancangan pangkalan data menggunakan metode *bottom-up*. Tujuan dari perancangan ini untuk melayani informasi strategis seperti jumlah total Mahasiswa baru per batch, status mahasiswa, dan jumlah hasil berdasarkan PGA dan predikat. Hasil dari perancangan ini berupa pemodelan skema bintang yang menjadikan data lebih teroganisir, memudahkan, dan proses query analisis lebih sederhana untuk penyajian data sebagai mendukung pelaporan dan proses evaluasi.

Arik Sofan Tohir beserta kedua temannya Kusrini dan Sudarmawan (2017), melakukan perancangan yang berjudul “Perancangan Data Warehouse Alumni Untuk Mendukung Kebutuhan Informasi Business Placement Centre Universitas AMIKOM”. dalam perancangan ini metode yang digunakan adalah metode Kimball dengan metodologi “nine-step methodology”, untuk perancangan pangkalan data menggunakan metode On-Line Analytical Processing dan Nine-steps method, tujuan menggunakan metode ini adalah dapat menampilkan data multi dimensi dengan cepat..Hasil dari perancangan ini didapatkan sebuah informasi fakta alumni, dimana data hasil proses Extract Transform and Load (ETL) data transaksional ditampilkan dalam bentuk pivot.

Akhmad Dahlan beserta kedua temannya Emma Utamidan Emha Taufiq Lutfhi (2017), juga melakukan perancangan yang berkaitan dengan data pangkalan data, penilaian ini berjudul “Perancangan Data Warehouse Perpustakaan Perguruan Tinggi XYZ Menggunakan Metode Snowflake Schema” Metode yang digunakan dalam penelitian ini adalah metode snowflake schema dan proses ETL untuk mengintegrasikan, mengekstraksi, membersihkan, mentransformasi serta mempopulasikan ke dalam data pangkalan data. tujuan dari perancangan dengan metode snowflake scema ini digunakan untuk mengakomodasi lebih banyak tabel dimensi karena nantinya akan memiliki sub-tabel dimensi lain sehingga bisa menghasilkan lebih banyak informasi yang akan dijadikan sebagai bahan untuk mengambil keputusan. Hasil dari perancangan ini berupa perancangan pemodelan snowflake scema dari data perpustakaan, penghilangan noise data yang kemudian terbentuk menjadi *reporting* yang sesuai dengan bussines requiremet, dari laporan ini dapat mengetahui tentang laporan jumlah peinjaman pertahun, laporan buku terlaris, laporan haru paling banyak peminjam, laporan pengarang terlaris, dan laporan penerbit terlaris.

Umi Fadilah, Wing Wahyu Winarno dan Armadyah Amborowati (2016) melakukan Perancangan pangkalan data yang berjudul “PERANCANGAN DATA WAREHOUSE UNTUK SISTEM AKADEMIK STMIK KADIRI” Metode yang digunakan adalah metode penelitian tindakan menggunakan action research

diantaranya tahap Observe, taap Reflect, Plan dan Act.. Hasil dari penelitian ini adalah Rancangan pangkalan data akademik yang sesuai dengan STMIK Kadiri dan menghasilkan laporan tentang jumlah mahasiswa per angkatan, per jenis kelamin, dan per kota asal mahasiswa.

Penelitian lain yang berkaitan dengan pangkalan data juga dilakukan oleh Aulia Paramita (2016) yang berjudul “Perancangan *Data Warehouse* pada Perpustakaan Yayasan Lentera Insan”. Didalam penelitian ini, peneliti mengembangkan pangkalan data pada perpustakaan dimana pada sumbernya masih menggunakan cara yang manual untuk menghubungkan tiap modul-modul yang terpisah. Kebutuhan untuk dilakukan penelitian ini disesuaikan dengan kebutuhan manajemen perpustakaan, penelitian ini mempunyai hasil data dimana data tersebut menjadi lebih terstruktur dan terintegrasi sehingga memudahkan pihak perpustakaan dalam mengambil keputusan.

Menurut Dinda Wilanda (2016) penelitiannya yang berjudul “Pembangunan *Data Warehouse* pada PT.PUPUK ISKANDAR MUDA”. Peneliti menggunakan metode pemodelan skema *Fact constellations* dan perancangan *OLAP(Online Analytical Processing)*, sedangkan untuk pengolahan dan analisis datanya peneliti menggunakan metode dengan teknik *Roll-Up* dan *Drill-Down*. Hasil dari penelitian menunjukkan bahwa metode pemodelan skema *Fact constellations* dan perancangan *OLAP* akurat diterapkan untuk mengintegrasikan dan menganalisa data penjualan dan produksi perusahaan tujuannya adalah untuk membantu perusahaan dalam mengintegrasikan data dan menganalisa penjualan dan produksi barang

Penelitian lain yang berkaitan dengan pangkalan data berjudul “Pengisian Borang Akreditasi Standar 3 Menggunakan Model *Data Warehouse*” oleh Deppi Linda (2016). Penelitian ini dilakukan dalam rangka untuk mendukung pengisian borang akreditasi terutama pengisian formulir standar 3 mengenai data kemahasiswaan dan lulusan serta mendapatkan informasi yang berkaitan dengan jumlah calon mahasiswa, jumlah mahasiswa baru pada fakultas, jumlah lulusan yang sekaligus dikelompokkan berdasarkan jenis indeks prestasi kumulatif beserta persentasinya. Penelitian ini melakukan perancangan pangkalan data dengan

metodologi *analisis Online Analytical Processing* dan metode pengembangan *Business Life Cycle*.

Penelitian yang dilakukan oleh Ramanti Dharayani, Kusuma Ayu Laksitowening dan Amarilis (2015) yang diberi judul penelitian “Implementasi ETL(Extract, Transform, Load) Pangkalan Data Perguruan Tinggi dengan Menggunakan State Space Problem”. Pada penelitian ini mengimplementasikan ETL dengan menggunakan alur kerja state space problem. State Space Problem sendiri digunakan untuk menggambarkan tahapan alur proses ETL dan mencari urutan aktivitas pada proses ETL. Dari hasil pengujian ETL dilakukan perubahan urutan aktivitas dengan menggunakan transisi graf dan didapatkan hasil yang lebih optimal.

Setiawan, Noor Akhmad, dan Sri Suning Kusumawardani (2014) melakukan perancangan pangkalan data yang berjudul “Perancangan Data Warehouse Sistem Informasi Eksekutif dengan Metode Kimbal (Kasus pada Universitas Muhammadiyah Magelang). Perancangan ini menggunakan metode 9 Langkah (Nine Step Methodology) dari Kimball, Tujuan menggunakan metode ini adalah karena memiliki tahapan perancangan yang dimulai dari kebutuhan, sehingga sistem mampu menyajikan informasi dengan struktur lengkap dan sesuai dengan kebutuhan pengambil keputusan. Hasil perancangan berupa pangkalan data yang telah digunakan untuk membangun salah satu bagian Sistem Informasi Eksekutif yaitu bidang Akademik (SIE-Akademik), Sistem ini mampu menyajikan data yang terintegrasi dan sapat ditampilkan sesuai dimensi faktual yang diinginkan berupa ringkasan IP mahasiswa selama 7 tahun.

Ni Wayan Wisswani (2013) juga melakukan penelitian yang berkaitan dengan pangkalan data yang berjudul “Penerapan *Hybrid Slowly Change Dimension* untuk *Nearly Realtime Data Warehouse*” Ni Wayan Wisswani menilai bahwa dalam melakukan perancangan pangkalan data menggunakan Teknik Pemodelan dimensi *Hybrid Slowly Change Dimension* dapat menangkap setiap perubahan data yang dapat mempengaruhi dimensi yang terjadi pada database sumber, karena Ni Wayan Wisswani juga mengembangkan teknik ini

menggunakan metode *change* dan *capture*. Sehingga penelitian ini mempunyai hasil terbentuknya desain SCD yang diimplementasikan dalam tabel-tabel form sistem thesis dan sistem disertasi yang berfungsi mencatat perubahan yang terjadi dalam sebuah kolom baru pada tabel, nilai kolom lama yang mengalami perubahan akan digunakan untuk menampung *field* awal sebelum perubahan dan kolom baru akan diisi nilai dari perubahan yang dilakukan.

Berdasarkan penelitian yang telah dilakukan beberapa orang diatas, ada beberapa penelitian yang berkaitan dengan pembangunan pangkalan data dengan menggunakan arsitektur dan cara yang berbeda-beda. Untuk itu yang menjadi fokus pada penelitian ini adalah bagaimana mengimplementasikan pembangunan pangkalan data alumni pelacakan studi Universitas Muhammadiyah Yogyakarta menggunakan arsitektur *normalized data store* dengan hasil akhir pelaporan dalam bentuk *dashboard*. Kelebihan arsitektur NDS ini adalah dalam pembangunannya dilakukan secara bertahap tanpa mengubah data asli, karena data dibentuk dalam 3 data *store*, dan hasil akhir dengan membuat pelaporan pangkalan data tersebut menggunakan *dashboard power bi* yang dipublikasikan, pelaporan ini mempermudah pemangku kebutuhan untuk menganalisis data alumni pelacakan studi tersebut karena dalam pembuatan pelaporan sudah terbuat secara sistematis.

## 2.2 Pengertian Data

Menurut bahasa Data diambil dari kata “*Datum*”, *Datum* sendiri berasal dari Bahasa Romawi yang bisa diartikan sebagai objek yang diberikan, oleh karena itu Data lebih untuk diberikan (fakta) bukan untuk memberikan, jika dikatakan sebagai memberikan maka data tersebut menjadi informasi yang telah baku dan dapat diketahui kebenarannya (Ramadhan, t.t.)

Data sangat erat dan lebih banyak ditemukan dalam istilah atau bagian komputer. Data juga dapat dikaitkan dengan database atau *software* pengolah data. Data bisa berbentuk dalam wujud yang banyak, contohnya seperti gambar, suara, angka, huruf, simbol-simbol, bahasa maupun suatu keadaan yang dapat digunakan sebagai bahan untuk melihat obyek yang luas maupun tertentu.

Data terdiri dari beberapa jenis, yang pertama adalah menurut sumbernya yaitu data primer dan sekunder, yang dimaksud oleh data primer adalah data baru atau asli, yang diperoleh dari sumber yang didapat dari objek penelitiannya sendiri atau terjun langsung ke lapangan. Sedangkan data sekunder adalah data tambahan, ditemukan dari sumber yang sudah ada atau terdahulu seperti jurnal, majalah, buku dan lain-lain

Jenis data yang lainnya adalah berdasarkan sifat-sifatnya, yaitu data kualitatif dan data kuantitatif. Data kualitatif banyak ditemui dalam bentuk pernyataan verbal atau suatu katakata, gambar, dan simbol tidak dijabarkan sebagai bentuk angka, cara mendapatkannya juga beragam, bisa dengan cara wawancara, observasi, ataupun diskusi. Sedangkan data kuantitatif adalah berbentuk angka atau secara terbilang, pengolahannya dengan menggunakan perhitungan matematika.

## **2.2 Pengertian *Database***

*Database* didefinisikan sebagai kumpulan data yang terorganisir yang disimpan dan diakses secara sistematis melalui media elektronik (pada umumnya komputer) yang dapat diubah maupun diolah menggunakan perangkat lunak (program aplikasi), sehingga dapat memberikan informasi yang optimal dan sesuai dengan yang dibutuhkan oleh pengguna (Tamba, 2018)

*Database* terdiri dari kumpulan tabel-tabel yang menyimpan data serta informasi, database juga merupakan peran yang sangat penting dalam sistem informasi, karena database berfungsi sebagai pusat untuk menyimpan data dari tahun-tahun sebelumnya hingga saat ini, yang nantinya akan diolah lebih lanjut.

Jadi secara konsep dapat disimpulkan bahwa database merupakan kumpulan dari data, berisi tabel-tabel yang membentuk suatu berkas dan data tersebut saling berelasi dengan tatacara tertentu untuk menghasilkan informasi sesuai dengan yang dibutuhkan oleh pengguna.

*Database* terdiri dari data yang digunakan oleh banyak pengguna, masing-masing pengguna sudah pasti memiliki kebutuhan dan keperluan yang berbeda-

beda, oleh karena sebab itu Sistem manajemen database menyediakan beberapa jenis, fungsi, dan kegunaan sesuai apa yang dibutuhkan oleh pengguna tersebut, berikut adalah tiga dari banyaknya jenis-jenis *database*, antara lain adalah :

1. *Operational Database*

*Database* jenis ini digunakan untuk mengelola data dinamis secara *real-time*, karena dapat menyimpan data secara lebih rinci, pengguna juga dapat memodifikasi data sesuai dengan apa yang diinginkan

2. *Relational Database*

*Database* jenis ini didasarkan menggunakan data *relational*, pengguna bisa dengan mudah mencari atau mengakses informasi dalam tabel-tabel yang berbeda. Hampir semua sistemnya menggunakan SQL (*Structured Query Language*) sebagai *query*nya.

3. *Distributed Database*

*Database* yang terdistribusi secara tersebar yang disimpan diberbagai perangkat komputer namun masih saling berhubungan antara satu sama lainnya. Dapat berada pada server jaringan yang terdesentralisasi di internet, intranet, ataupun ekstranet. Biasanya *database* jenis ini banyak digunakan pada suatu perusahaan yang memiliki beberapa cabang, agar memudahkan perusahaan atau pengguna untuk mengakses data walaupun berada di berbeda tempat.

### **2.3 Pengertian OLTP (*Online Transaction Processing*)**

OLTP merupakan sistem yang berorientasi proses, data operasional dimana data yang terlibat dalam pengoperasian sistem tertentu dan hal tersebut ditandai dengan adanya transaksi online sederhana, seperti *INSERT*, *UPDATE*, dan *DELETE*, memproses suatu transaksi secara langsung melalui sistem elektronik atau komputer yang terhubung dalam jaringan, contohnya seperti database toko atau swalayan pada sistem transaksi penjualan (Parekh, 2013)

Data dalam sistem OLTP sering diperbarui, hal ini bertujuan untuk mencegah redundansi data dan upaya untuk mencegah pembaruan anomali tabel

data dinormalisasi, hal tersebut yang membuat penulisan operasi pada tabel didalam database menjadi lebih efisien. OLTP banyak digunakan oleh pengguna IT Professional, OLTP merupakan suatu sistem berorientasi proses, bentuk databasenya harus dinormalisasi terlebih dahulu sebelum digunakan, karena data yang digunakan di OLTP digunakan setiap saat.

## **2.4 Pengertian OLAP (*Online Analytical Processing*)**

OLAP seringkali disebut dengan analisis data multidimensi, data multidimensi adalah data yang dapat dimodelkan sebagai atribut dimensi dan atribut ukuran, OLAP merupakan jenis perangkat lunak yang melakukan suatu proses yang digunakan untuk melakukan permintaan terhadap data dalam bentuk yang kompleks dan untuk menganalisa data bervolume besar dari berbagai perspektif (Parekh, 2013)

OLAP bekerja dengan data yang berbentuk multidimensi yang biasanya berbentuk 3 dimensi yang seringkali disebut dengan kubus (*cube*), *cube* merupakan bagian utama dari OLAP. *Cube* berisi kumpulan data-data yang sebelumnya telah disatukan sehingga dapat dengan cepat menemukan hasil *query*. OLAP juga merupakan suatu system yang berorientasi subjek, fungsinya sendiri lebih diutamakan dalam mendukung pengambilan keputusan dalam pangkalan data, sehingga data-data OLAP tidak bisa diganti, diedit maupun dihapus.

## **2.4 Konsep Data Pangkalan Data**

### **2.4.1 Pengertian Pangkalan Data**

Pangkalan data adalah tempat dimana menyimpan data historikal yang telah dibeda-bedakan berdasarkan pada subjek sebagai pendukung untuk pengambilan keputusan suatu analisa atau organisasi (Malik, 2005) Pangkalan Data adalah koleksi data yang bersifat analisis, digunakan untuk dijadikan inti dari sebuah pengambilan keputusan, pangkalan data bisa dikatakan sebagai database relasional karena desain dari pangkalan data mengacu pada *query* dan analisa dari proses transaksi, jadi pangkalan data memudahkan kerja

analisis untuk menyusun metode perancangan database, karena pangkalan data sendiri berisi tentang data *history* yang dipakai dalam proses analisis tersebut. Fokus dari database itu sendiri lebih ke pemodelan data dan desain data.

#### 2.4.2 Karakteristik Pangkalan Data

Pangkalan data memiliki karakteristik yang membuat pangkalan data berbeda dari database yang lainnya, terdapat empat karakteristik yaitu Berorientasi kepada subjek (*Subjek-Oriented*), Data yang terintegrasi (*Data Integrated*), Pembuatan dalam rentang waktu tertentu (*Timeline*), Data yang disimpan bersifat tetap (*Non-Volatile*) (Malik, 2005), keempat karakteristik tersebut akan dijelaskan sebagai berikut :

a. *Subject Oriented* (Berorientasi kepada subjek)

Pangkalan Data berorientasi Subjek adalah Pangkalan data yang dirancang untuk menganalisa data organisasi berdasarkan subjek-subjek tertentu, digunakan sebagai pengambilan keputusan dan menganalisis yang berhubungan dengan histori subjek-subjek utama tersebut. Tetapi bukan berdasarkan pada fungsi maupun proses aplikasinya, karena pangkalan data dijadikan sebagai penyimpanan data untuk mengambil keputusan, bukan untuk aplikasi berorientasi data.

b. *Data Integrated* (Data yang terintegrasi)

Pangkalan Data terintegrated adalah pangkalan data yang bisa menyimpan banyak data, berasal dari banyak sumber yang berbeda-beda atau terpisah pada suatu format yang bersifat konsisten namun dapat terintegrasi dari satu sama lainnya, dengan demikian dalam data tidak ada kerancuan data bahkan data tidak dapat rusak mapun dipisah-pisah karena data tersebut sudah menjadi satu kesatuan yang mendukung dari konsep keseluruhan pangkalan data itu sendiri.

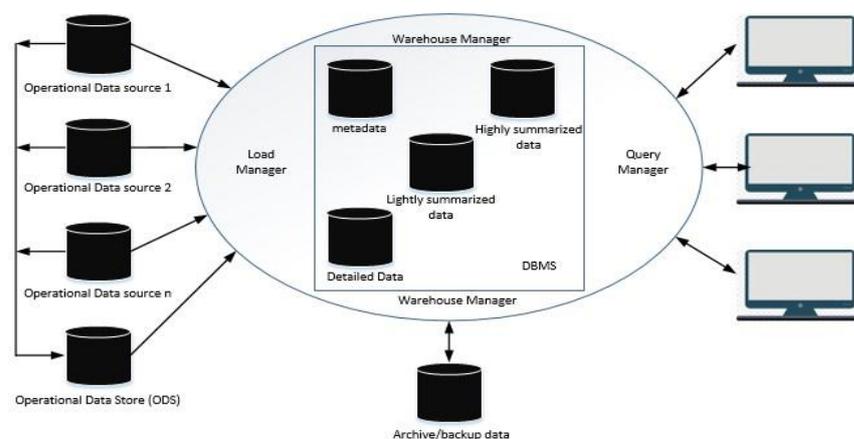
c. *Timeline* (Pembuatan dalam rentan waktu tertentu)

Pangkalan Data *Timeline* adalah pangkalan data yang mengandung data keseluruhan, dari yang bernilai sekarang sampai yang bernilai historis, data tersebut nantinya yang akan dibuat untuk dijadikan dalam pengambilan keputusan atau analisis, cara yang paling sederhana adalah jika akan menyajikan Pangkalan data pada rentan waktu yang tertentu, misalnya harian, mingguan, bulanan, bahkan tahunan, sebagai contoh antara 1 sampai 10 tahun kedepan.

d. *Non-Volatile* ( Data yang disimpan bersifat tetap)

Pangkalan Data *Non-Volatile* adalah data yang ada pada pangkalan data tersebut tidak di perbarui secara *real-time* tetapi di *refresh* dari sistem operasinya secara teratur, data yang ada dalam database operasional akan secara berkala dipindahkan ke dalam pangkalan data sesuai dengan jadwal yang sudah di tentukan, didalam pembangunan pangkalan data disini hanya terdapat dua kegiatan manipulasi, yaitu *access* data (mengakses pangkalan data seperti melakukan query/ menampilkan laporan yang dibutuhkan, disini tidak terdapat kegiatan perbarui datanya) dan *loading* data (menggambil data).

### 2.4.3 Arsitektur Pangkalan Data



**Gambar 2.1** Arsitektur Pangkalan Data (Connolly dan Begg, 2010:1204)

Komponen-komponen yang ada di dalam arsitektur pangkalan data seperti gambar 2.1 diatas terbagi menjadi beberapa bagian, yaitu :

a. *Operational Data*

*Operational Data* adalah data yang digunakan untuk proses operasional bisnis dalam jangka waktu atau sehari-hari.

b. *Operational Data Store (ODS)*

*Operational Data Store* adalah wadah atau tempat digunakan sebagai penyimpanan data operational yang terintegrasi, dapat digunakan untuk menganalisis data untuk mendukung segala proses analisis maupun proses transaksi operational, ODS juga mempermudah kinerja pembangunan pangkalan data karena dapat menyediakan data yang sudah di ekstrak dan di bersihkan dari pusat atau sumbernya, sehingga untuk proses pengintegrasian dan restrukturisasi datanya menjadi lebih sederhana.

c. *Load Manager*

*Load Manager* bisa disebut juga sebagai front-end yang bertanggung jawab untuk menangani semua operasi yang berhubungan dengan ekstrasi dan load data (menaruh data) ke dalam pangkalan data.

d. *Warehouse Manager*

*Warehouse manager* bertanggung jawab atas semua operasi yang berhubungan dengan pengelolaan data di dalam pangkalan data, bertugas untuk menganalisis data agar menjaga konsistensi data, *warehouse manager* juga melakukan transformasi dan penggabungan sumber data dari tempat penyimpanan yang sementara dan dimasukkan ke dalam tabel-tabel pangkalan data tersebut.

e. *Query Manager*

*Query Manager* juga bisa disebut dengan komponen backend, berfungsi untuk menampilkan segala operasi yang terkait dengan *query* pengguna, meliputi kegiatan yang menunjukkan jalan atas permintaan ke tabel-tabel data yang tepat.

f. *Detailed Data*

Dalam Pangkalan data area ini berfungsi sebagai tempat penyimpanan semua detailed data dalam schema database, detailed data dibagi menjadi dua bagian yaitu current detail (tempat menyimpan semua detailed data yang baru) dan old detailed (tempat menyimpan semua detailed data yang lama).

g. *Lightly and Highly Summarized Data*

*Lightly dan Highlly Sumarrized Data* adalah tempat untuk penyimpanan yang sementara data predefinisi yang ringkas secara *light and high* yang sebelumnya dibuat oleh *warehouse manager*. Tujuan dari ringkasan informasi ini adalah untuk mempercepat tanggapan terhadap permintaan user. Ringkasan data ini diperbarui secara berkala dengan bertambahnya data dalam pangkalan data.

h. *Archive or Backup Data*

Dalam pangkalan data area ini digunakan untuk menyimpan semua detail dan data yang sudah diringkas. Tujuannya untuk *archiving* dan *backup* data yang kemudian dikirim ke media penyimpanan seperti *magnetig tape* atau *optical disk*.

i. *Metadata*

*Metadata* adalah data yang mencakup data yang mendeskripsikan pangkalan data, berfungsi sebagai tempat penyimpanan data dari semua definisi metadata yang kemudian digunakan dalam seluruh proses pembuatan pangkalan data, metadata digunakan dengan tujuan untuk proses proses *warehouse management*, sebagian proses *query management*, *extracting* dan *loading*.

j. *End User Access Tools*

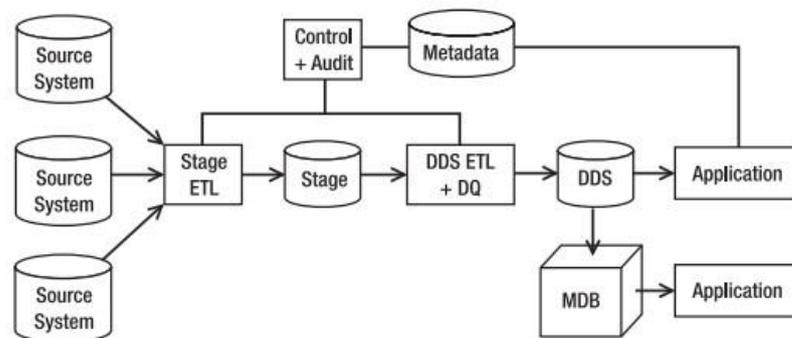
*End User Access Tools* merupakan *tools* yang memanfaatkan kegunaan dari pangkalan data, menyediakan data yang konsisten kepada pengguna yang nantinya digunakan untuk menganalisis informasi dalam proses informasi eksekutif, pengambilan keputusan, OLAP, pembuatan laporan, dan data mining.

#### 2.4.4 Data Flow Arsitektur Data warehouse (Arsitektur Aliran Pangkalan Data)

Dalam Pangkalan Data atau pangkalan data terdapat empat *data flow* arsitektur, yaitu single DDS, NDS + DDS, ODS + DDS dan FDW (*Federated Data Warehouse*), *Data Flow* Arsitektur ini digunakan untuk memutuskan waktu untuk membuat pembangunan sistem Pangkalan Data , berikut penjelasannya :

- Single DDS

Arsitektur Single DDS yang ada pada gambar 2.2 ini memiliki satu dimensional data *store*, Dalam arsitektur ini pengguna memiliki penyimpanan data pada satu dimensi, DDS terdiri dari satu atau beberapa *dimensional data mart*, sebuah ETL (*extract, transform, and load*) mengekstrak data dari berbeda sumber dan memindahkannya pada Stage. Stage adalah tempat dimana pengguna menyimpan data yang sudah di ekstrak dari sistem sumber untu sementara, sebelum di proses lebih jauh lagi.

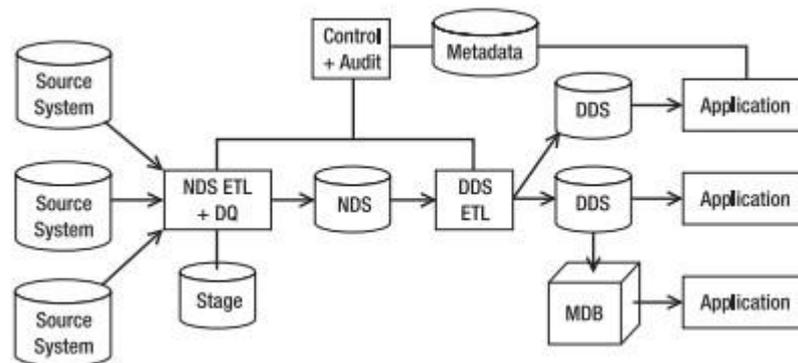


**Gambar 2.2** Arsitektur *Single DDS* (Vincent Rainardi, 2008)

Keuntungan menggunakan arsitektur ini adalah lebih simpel dibandingkan dengan arsitektur yang lainnya, karena data dari stage akan dimuat langsung kedalam *dimensional data store* (DDS) tanpa haru melalui normalisasi *data store* terlebih dahulu, namun kekurangannya adalah untuk membuat DDS kedua sangat sulit.

- NDS + DDS

Didalam arsitektur yang ada pada gambar 2.3 ini terdapat tiga *data store* yaitu stage, NDS dan DDS. Arsitektur ini mempunyai kemiripan dengan single DDS, bedanya hanya pada arsitektur ini terdapat normalisasi data terlebih dahulu sebelum dimuat kedalam DDS, Tujuannya adalah NDS dapat mengintegrasikan data dari beberapa sumber sistem.



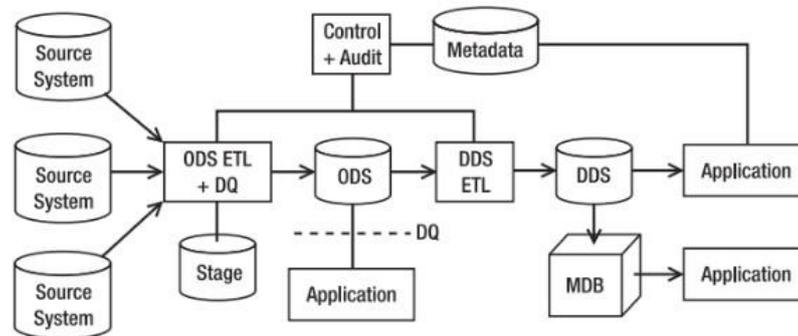
**Gambar 2.3** Arsitektur data NDS + DDS (Rainardi, 2008)

Keuntungan dari NDS + DDS ini adalah pembuatan DDS akan sangat mudah, karena NDS berjalan sebagai master dari *data store* yang berisi tentang set data yang lengkap. Hal ini bisa dijadikan untuk membuat *static data store* yang terpisah dengan tujuan untuk melakukan analisis. Namun kerugiannya adalah dibutuhkan usaha yang lebih ekstra dibandingkan dengan arsitektur single DDS karena data dari sumber yang dimasukkan ke dalam stage perlu dimuat ke dalam NDS terlebih dahulu sebelum nantinya dimuat ke dalam DDS dan selanjutnya dibutuhkan dua proses ETL dalam arsitektur ini.

- ODS + DDS

Arsitektur pada gambar 2.4 ini hampir mirip dengan NDS + DDS bedanya hanya NDS digantikan dengan ODS. ODS mengintegrasikan data dari berbagai sistem sumber. ODS hanya berisi versi master data.

Data dalam ODS diersihkan terlebih dahulu dan diintegrasikan. Data yang mengalir ke dalam ODS telah melewati *screening DQ*.



**Gambar 2.4** Arsitektur data ODS + DDS (Rainardi, 2008)

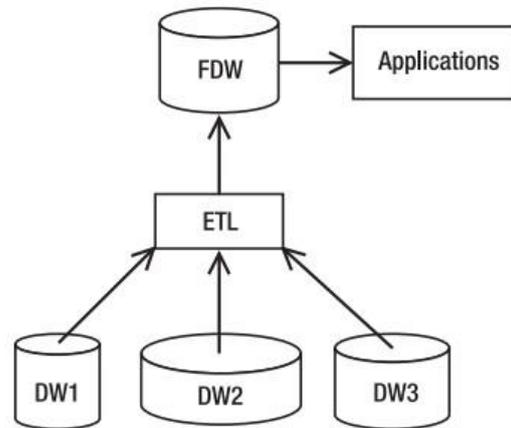
Keuntungan dari arsitektur ini adalah bentuk normal ketiga lebih slm daripada NDS karena dalam arsitektur ini berisi nilai saat ini. Hal ini membuat performa dari ODS ETL dan DDS ETL lebih bagus daripada performa pada arsitektur NDS + DDS, didalam arsitektur ini *Normalized relational* diupdate oleh aplikasi *end-user*. Sedangkan kerugian dari arsitektur ini adalah ketika akan membuat DDS, pengguna diharuskan untu mendapatkan dari main DDS dan tidak bisa menggunakan DDS ETL yang ada.

- FDW ( *Federated Data Warehouse* ).

Arsitektur yang ada pada gambar 2.5 ini berisi beberapa Pangkalan Data dengan data *retrieval layer* diatasnya, pengambilan data dari Pangkalan Data yang ada menggunakan ETL dan memuatnya kedalam DDS yang baru.

Keuntungan menggunakan arsitektur ini adalah kemampuan untuk mengakomodasi Pangkalan Data yang ada maka dari itu waktu yang dibutuhkan development akan lebih cepat. Sedangkan kerugian dari arsitektur ini adalah sulitnya untu membuat kualitas data yang

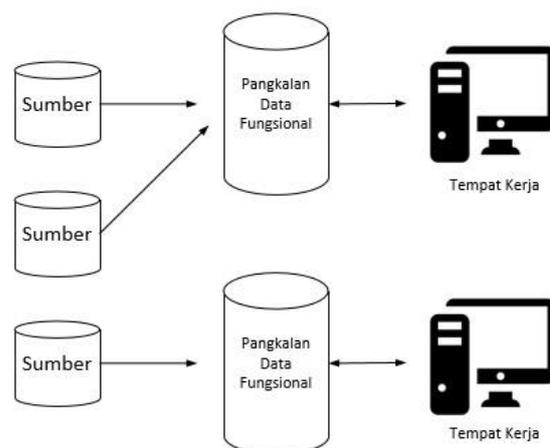
bagus dari Pangkalan Data sebelumnya yang belum mempunyai standarisasi.



**Gambar 2.5** Arsitektur Aliran Data *Federated Data Warehouse* (Vincent Rainardi, 2008)

#### 2.4.5 Dasar Sistem Pangkalan Data

Dalam Pangkalan Data ada tiga jenis Dasar sistem, yaitu *Funtional Data Warehouse* (Pangkalan Data Fungsional), *Data warehouse* (Pangkalan Data Terpusat), *Distributed Data Warehouse* (Pangkalan Data Terdistribusi) berikut penjelasannya :



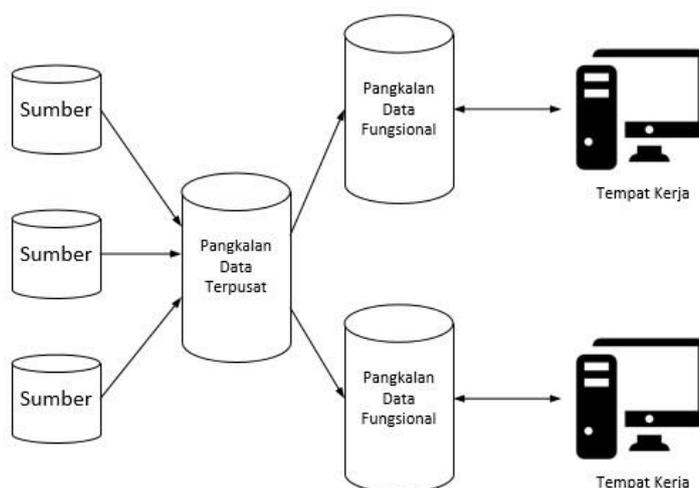
**Gambar 2.6** Sistem *Funtional* Pangkalan Data

1. *Funtional Data Warehouse* (Pangkalan Data Fungsional)

Pangkalan Data dibuat lebih dari satu dan dikelompokkan berdasarkan fungsi-fungsi yang ada dalam organisasi, kata operasional disini merupakan *database* yang diperoleh dari kegiatan sehari-hari. Seperti yang dijelaskan pada Gambar 2.6 diatas, Keuntungan bentuk pangkalan data ini adalah sistem yang mudah dibangun dengan biaya yang cukup relatif murah, sedangkan kerugiannya adalah resiko untuk kehilangan konsistensi data dan kemampuan yang terbatas dalam pengumpulan datanya bagi pengguna.

## 2. *Datawarehouse* (Data Warehouse Terpusat)

Pada gambar 2.7 menunjukan bahwa bentuk yang terlihat seperti bentuk *data warehouse fungsional*, biasanya pangkalan data terpusat ini biasa digunakan oleh organisasi yang belum memiliki jaringan eksternal, keuntungan bentuk pangkalan data ini adalah datanya benar-benar terpadu karena konsistensinya yang tinggi, namun kerugiannya adalah biaya yang dikeluarkan relatif mahal serta memerlukan waktu yang lama untuk pembangunannya.

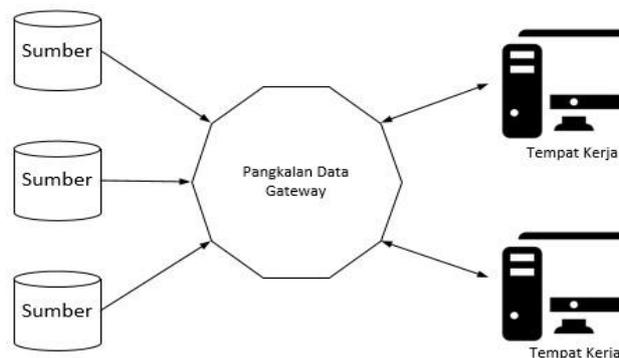


**Gambar 2.7** Sistem Pangkalan Data terpusat

Pada gambar 2.7 diatas menunjukan bahwa bentuk yang terlihat seperti bentuk data warehouse fungsional, biasanya pangkalan data terpusat ini biasa digunakan oleh organisasi yang belum memiliki jaringan eksternal, keuntungan bentuk pangkalan data ini adalah datanya

benar-benar terpadu karena konsistensinya yang tinggi, namun kerugiannya adalah biaya yang dikeluarkan relatif mahal serta memerlukan waktu yang lama untuk pembangunannya.

### 3. Distributed Data Warehouse (Pangkalan Data *Terdistribusi*)



**Gambar 2.8** Sistem Pangkalan Data Terdistribusi

Pada gambar 2.8 di atas ditunjukkan *gateway* berfungsi sebagai jembatan penghubung antara pangkalan data dengan *workstation* (Tempat Kerja) yang menggunakan banyak sistem. Dengan sistem terdistribusi ini memungkinkan organisasi dapat dengan mudah mengakses sumber data yang berada pada akses eksternal (diluar organisasi), keuntungan bentuk pangkalan data ini adalah datanya konsisten terus menerus karena sebelum datanya digunakan, data terlebih dahulu mengalami proses sinkronisasi, namun kerugiannya adalah lebih kompleks atau sulit untuk diterapkan, karena sistem operasinya dikerjakan secara terpisah dan biayanya paling mahal dibandingkan dengan dua bentuk pangkalan data yang lainnya.

## 2.5 Model Dimensional

Model Dimensional adalah sebuah teknik dalam melakukan perancangan logikal yang mempunyai tujuan untuk mempresentasikan data kedalam sebuah standar, bentuk yang intuitif dan dapat diakses walaupun dengan performa dan akses kerja yang tinggi (Connolly dan Begg, 2005).

Model dimensional memungkinkan pengguna dalam mengorganisir tabel-tabel menjadi lebih praktis, karena dapat mengakomodasi kompleksitas dari basis datanya sendiri dan dapat membuat peningkatan kinerja pengguna. Model dimensional menggunakan konsep hubungan antar *entity* (ER) dengan beberapa batasan yang penting, setiap model dimensinya terdiri dari satu tabel dengan *multipart key* yang biasa disebut dengan tabel fakta dan dengan sebuah set tabel yang lebih kecil yaitu tabel dimensi. Setiap tabel dimensi memiliki sebuah *primary key* yang sama persis dengan salah satu dari komponen *multipart key* dalam tabel fakta.

Fitur yang sangat penting dalam Model Dimensional adalah semua *natural keys* diganti dengan kunci pengganti (*surrogate keys*), oleh karena itu setiap melakukan fungsi *join* antar tabel dengan tabel dimensi yang lainnya harus didasari dengan kunci pengganti, kegunaannya adalah memperbolehkan data yang ada pada Pangkalan Data agar memiliki kebebasan dalam penggunaan datanya.

Di dalam Model Dimensional terdapat beberapa istilah, yaitu tabel fakta dan tabel dimensi, penjelasannya adalah sebagai berikut :

- Tabel Fakta (*Fact Table*)

Pada umumnya tabel fakta mengandung sesuatu yang bisa diukur (*measure*), contohnya seperti harga, jumlah barang, dan lain-lain. Tabel fakta merupakan kumpulan dari *foreign key* dari *primary key* yang terdapat pada masing-masing dimension table, Tabel fakta juga mengandung data yang bersifat lama atau historis.

- Tabel Dimensi

Tabel Dimensi adalah tabel yang berisi data yang mendetail, menjelaskan *foreign key* yang terdapat pada tabel fakta. Atribut-atribut yang ada pada tabel dimensi dibuat secara hirarki (berjenjang) agar memudahkan saat proses *query*.

Ada beberapa Tabel dimensi yang seringkali digunakan dalam pembuatan Pangkalan Data, diantaranya adalah sebagai berikut :

### 2.5.1. *Star Schema* (Skema Bintang)

Skema bintang adalah model arsitektur dari Pangkalan Data yang paling sederhana, disebut skema bintang karena bentuk diagramnya menyerupai bintang, dengan titik-titik yang memancar dari pusatnya (*center*). Pusat bintang terdiri dari tabel fakta dan titiktitiknya adalah tabel dimensi, biasanya tabel fakta dalam skema bintang berada pada bentuk normal ketiga (3NF), sedangkan tabel dimensi harus dinormalisasi terlebih dahulu, Terlepas dari itu kenyataannya adalah bahwa skema bintang adalah arsitektur yang paling sederhana dibandingkan dengan arsitektur yang lainnya, dan skema bintang banyak digunakan oleh beberapa pengguna (Kimball & Ross, 2011).

Kelebihan menggunakan skema bintang adalah skema bintang cenderung lebih mudah dipahami karena bentuk modelnya yang sederhana, dengan begitu lebih memudahkan untuk mencari isi dengan cara melihat satu persatu dari setiap masing-masing dimensinya, Proses *query* lebih cepat pada saat melakukan proses OLAP, namun kekurangannya adalah ukuran datanya lebih besar dibandingkan yang lainnya, karena ada data yang disimpan ulang, untuk update dan maintenance juga lebih sulit.

### 2.5.2 *Snowflake Schema*

*Snowflake schema* menyerupai bentuk *star schema*, mengacu pada database multidimensi dengan tabel logical, dimana *entity-relationship* diagram disusun menjadi bentuk *snowflake*, hal ini sering digambarkan oleh tabel fakta terpusat yang terkait dengan tabel dimensi, tabel dimensi dibagi ke dalam berbagai tabel-tabel dimensi yang lainnya, yang umumnya dinormalisasi untuk menghapus redundansi (duplikat data), Misalnya tabel produk dibagi menjadi beberapa tabel produk dan sub-produk. *Snowflake schema* dibuat dalam permintaan yang fleksibel atas hubungan antar dimensi yang lebih rumit, hal ini sangat ideal untuk hubungan dari *one-to-many* dan *many-to-many* diantara tingkat dimensi, dan biasanya terkait dengan data mart maupun dimensi pada Pangkalan Data (Kimball & Ross, 2011).

Kelebihan menggunakan *snowflake schema* ini adalah ukuran datanya cenderung lebih kecil di dalam tempat penyimpanan, proses *querynya* juga lebih cepat pada saat melakukan proses ETL, dan juga pada skema ini lebih mudah untuk dilakukan *update* dan *maintenance*. namun kekurangannya adalah pada *snowflake schema* ini lebih sulit untuk memahaminya, kerana bentuknya yang lebih kompleksitas, dan karena itu untuk mencari data yang dibutuhkan cenderung lebih sulit karena strukturnya yang kompleks dan bercabang-cabang.

### 2.5.3 *Starflake Schema*

*Starflake schema* merupakan gabungan dari kedua *schema*, yaitu *star schema* dan *snowflake schema*, keduanya merupakan model-model dimensional. *Starflake schema* adalah model data dimensional yang memiliki sebuah tabel fakta sebagai pusatnya dan dikelilingi oleh tabel-tabel dimensi yang ternormalisasi dan terdenormalisasi (Kimball & Ross, 2011).

keuntungan menggunakan model ini adalah lebih efisien dalam mengakses datanya, dapat dengan mudah untuk beradaptasi terhadap kebutuhan-kebutuhan setiap pengguna, dan sifatnya fleksibel terhadap perubahan yang terjadi khususnya dalam perubahan yang mengarah pada perkembangan.

## 2.6 *Extract, Load, and Transaction (ETL)*

ETL adalah sebuah proses untuk menghasilkan Pangkalan Data yang terdiri dari *extracting*, *transforming*, *loading*, dan beberapa proses yang perlu dilakukan sebelum dipublikasikan ke dalam Pangkalan Data. ETL adalah sistem untuk proses pencarian data dari satu *data store*, merubah bentuk data, pengintegrasian data, dan menyimpan ke *data store* yang lainnya, *Data store* yang telah dibaca ETL disebut juga dengan data source, sedangkan *data store* yang disimpan kedalam ETL disebut dengan target (Caserta, 2004) . Proses ini terdiri dari tiga tahap , yaitu:

### 2.6.1 *Extraction*

Ekstraksi data adalah proses dimana pengambilan data atau pengekstrakan data dari satu database atau beberapa database yang berbeda, baik menggunakan *query* atau aplikasi ETL. Proses ekstraksi mencakup tugas yaitu memvalidasi data dan erase (penghapusan) data agar mendapatkan suatu pola atau struktu data yang cocok atau diinginkan.

Ekstraksi data biasanya terjadi dalam salah satu dari tiga cara sebagai berikut :

- *Update notification* (Perbarui pemberitahuan)  
Sistem ini memberi tahu pengguna ketika catatan atau datanya diubah, biasanya disebut dengan *easiest method of extraction* (metode ekstraksi yang termudah)
- Incremental extraction (Ekstraksi tambahan)  
Beberapa sistem tidak dapat memberikan notifikasi untuk melakukan pembaruan, sehingga mereka mengidentifikasi ketika catatan atau data telah dimodifikasi dan memberika ekstraksi khusus pada hal tersebut.
- *Full extraction* ( Ekstrasi yang penuh)  
Beberapa sistem sama sekali tidak dapat mengidentifikasi kapan datanya telah diubah, sehingga satu-satunya cara untuk mengeluarkannya dari sitem adalah dengan *reload* (memuat ulang) semuanya, biasanya ini hanya disarankan untuk jumlah data yang kecil sebagai pilihan yang paling terakhir.

### 2.6.2 Transaction

Transformasi adalah proses mengubah data, dimana data yang masih mentah (raw data) dari hasil ekstrasi disaring dan dirubah sesuai dengan yang diinginkan, diangkut secara fisik kedalam target yang dituju. Transformasi data ini dapat mencakup operasi seperti *cleaning*, *joining*, dan *validating* data

dan menghasilkan data yang terhitung berdasarkan nilai yang ada (“ETL Transform,” t.t.).

Ada beberapa jenis transformasi umum dan lanjutan yang bekerja untuk menyiapkan data analisis, beberapa diantaranya adalah :

a. *Basic Transformations* (Transformasi dasar)

- *Cleaning* (Pembersihan)
- *Format revision* (Revisi format)
- *Restructuring* (Restrukturisasi)
- *Deduplication* (*Deduplikasi*)

b. *Advanced Transformations* (Transformasi lanjutan)

- *Joining* (Penggabungan)
- *Splitting* (Pemisahan)
- *Derivation* (Penurunan)
- *Summarization*
- *Integration* (*Integrasi*)

### 2.6.3 Load

Load merupakan tahapan terakhir dalam proses ETL, melibatkan pemuatan data yang didapatkan dari hasil transformasi diubah menjadi target akhir (“ETL Transform,” t.t.). Target ini bisa berupa menjadi database atau Pangkalan Data. Ada dua *primary method* (metode utama) untuk memuat data ke Pangkalan Data, yaitu :

- *Full load* (beban penuh)  
Melibatkan seluruh data dump yang terjadi saat sumber pertama muat ke dalam Pangkalan Data.
- *Incremental load* (*beban tambahan*)  
Terjadi secara berkala, Interval ini dapat berupa *streaming increments* (Penambahan arus) lebih baik digunakan untuk volume data yang lebih kecil