BAB II

LANDASAN TEORI

Pada bagian ini penulis menjelaskan landasan teori yang menjadi acuan dalam penelitian ini yaitu landasan teori mengenai *data warehouse* dan *data mining*. Pada sub bab *data warehouse* penulis menjelaskan pengertian *data warehouse*, model arsitektur *data warehouse* yang akan digunakan berikut teknik dan proses yang akan dilakukan untuk membentuknya. Pada sub data *mining* akan dijelaskan mengenai pengertian dan kegunaan *data mining* berikut teknik dan proses membuatnya.

2.1 *Data Warehouse*

Data warehouse merupakan *database relational* yang didesain untuk melakukan lebih dari sekedar proses transaksi tapi lebih mengarah pada proses query dan analisa. Biasanya terdiri dari *historical* data yang diambil dari *transaction* data, tapi bisa juga berasal dari sumber yang lain. *Data warehouse* memisahkan antara analisa dan transaksi dan memungkinkan suatu organisasi untuk mengkonsolidasikan data dari beberapa *source*.

2.1.1 Definisi *Data Warehouse*

Muntean (2007) mendefinisikan *data warehouse* sebagai kumpulan informasi yang disimpan dalam *database* yang digunakan untuk mendukung pengambilan keputusan dalam sebuah organisasi. Data dikumpulkan dari berbagai aplikasi yang telah ada. Data yang telah dikumpulkan tersebut kemudian *divalidasi* dan *direstrukturisasi* lagi, untuk selanjutnya disimpan dalam *data warehouse*. Pengumpulan data ini memungkinkan para pengambil keputusan untuk pergi hanya ke satu tempat untuk mengakses data yang ada tentang organisasinya.

Sebagai tambahan, data memuat proses *extraction, transportation, transformation, and loading (ETL) solution*, mesin *online analytical processing*

(OLAP), *client analysis tools*, dan aplikasi lainnya yang mengatur proses pengumpulan dan pengiriman data ke *user*.

Data warehouse sering menjadi bagian *inti dari infrastruktur business intelligent* (BI) organisasi. *Data warehouse* adalah kumpulan dari basis data yang terintegrasi dan *subject oriented* yang didesign untuk mendukung DSS (*Decision Support Systems*). Karakter utama dari *data warehouse* antara adalah lain (Muntean *et al*, 2007):

- *Subject oriented*, data disusun dan diorganisasikan berdasarkan bagaimana *users* menggunakannya.
- Semua sifat ketidak *konsistenan* yang disebabkan oleh kesepakatan penamaan dan representasi nama dihilangkan
- *Time Variant*, data bersifat kekinian tapi lebih bersifat *time series.*
- *Non volatile*, data di simpan dalam dalam format *read-only* dan tidak akan berubah.

2.1.2 Penelitian Implementasi *Data Warehouse*

Bentuk-bentuk penelitian dalam perancangan *data warehouse* meliputi berbagai bidang. *Data warehouse* dimanfaatkan untuk mendapatkan informasi kinerja dosen (jumlah mata kuliah yang diajarkan, jumlah kelulusan/ ketidakllusan), kinerja mahasiswa (jumlah mata kuliah yang lulus/tidak lulus dibanding mata kuliah yang diambil), summary tiap nilai mata kuliah yang memiliki nilai A, B, dan C, (Handojo *et al*, 2004). *Data warehouse* juga digunakan dibidang medis untuk membantu menyediakan sumber data dalam infrastruktur BI dunia medis. BI dalam mengambil keputusan dalam memberikan tindakan terbaik terhadap pasien (Bhattacharyya ,2005).

Menurut Bhattacharyya (2005), BI adalah serangkaian proses untuk mengubah data menjadi informasi yang pada akhirnya menjadi pengetahuan Data adalah angka-angka, gambar, kata-kata dan lain-lain. Data mendorong terbentuknya informasi atau pengetahuan. Aplikasi *operasional* menyimpan data transaksi secara simultan, sehingga pertambahan data tidak bisa dihindarkan.

Semakin besar data tersimpan, akan mempengaruhi performace aplikasi, terutama modul *reporting*.

Handojo dan Silivia (2004) melaporkan bahwa pada kasus implementasi *data warehouse* di Universitas Petra Surabaya, dengan adanya *data warehouse,* proses penyusunan laporan menjadi lebih sederhana, karena pengguna bisa melakukan *customization report* sesuai dengan yang diinginkan, sehingga tercipta efisiensi waktu dari yang sebelumnya satu bulan (dengan program tambahan) atau seminggu (manual) menjadi satu hari.

Ariana (2007) juga menyebutkan hal yang sama bahwa implementasi *data warehouse* dalam organisasinya (Universitas Nasional) membantu pengambil kebijakan. Pada kasus UNAS lebih spesifik digunakan untuk mengenali pola karakterisitik mahasiswa yang mengambil program peminatan tertentu di program studi Manajemen Perusahaan UNAS dengan dibantu *data mining.*

2.1.3 Sistem Operasional dan Sistem Pendukung Pengambilan Keputusan

Data operasional dan data yang tersimpan berbeda. Ponniah (2001) memaparkan perbedaan tersebut diantaranya;

1. Dalam sistem *operasional*, data yang disimpan menunjukan data yang sekarang (*Current Values*), sedangkan apa yang tersimpan dalam *data warehouse* adalah *data archived* atau data sebagai hasil penurunan-penurunan atau *summarized* dari suatu data besar.
2. Dilihat dari struktur keduanya juga berbeda, data operasional *didesign* dengan sedemikian sehingga optimum untuk melakukan transaksi, sedangkan dalam *data warehouse* dioptimalkan untuk menghandle *query* yang rumit.
3. Penggunaan data *operasional* bersifat perulangan, sedangkan *data warehouse* bersifat *ad-hoc*, acak dan *heuristic*.
4. Ditinjau dari *frequency* mengaksesnya, data operasional sangat sering diakses sedangkan *data warehouse* levelnya sedang atau jarang.
5. Jika dalam data *operasional* akses terhadap datanya bisa *read, update* atau bahkan *delete*, maka dalam *data warehouse* hanya bisa *read*.

6. Jumlah pengguna juga berbeda, jika operasional digunakan oleh orang banyak, sedangkan data warehouse oleh beberapa orang saja untuk mendukung analisa keputusan.

Perbedaan tersebut adalah karena memang adanya perbedaan tujuan dari dibuatnya sistem. Data Operasional digunakan untuk menjalankan operasional bisnis dengan cara memasukan data-data kedalamnya. Dalam data *warehouse* tujuannya adalah mendapatkan informasi yang bisa mendukung pengambilan keputusan. Pengguna bisa produk mana yang menjadi favorit, daerah mana yang banyak mengalami masalah penjualan, kenapa bisa terjadi masalah (*drill down*), yang pada prinsipnya untuk menangkap peluang dan mengurangi resiko dalam menjalankan bisnis.

2.1.4 Keuntungan Data Warehouse

Implementasi *data warehouse* yang tepat dapat memberikan keuntungan-keuntungan antara lain:

1. *Meningkatkan produktifitas dari pengambil keputusan perusahaan*

   Data warehouse meningkatkan produktifitas dari pengambil keputusan perusahaan dengan membuat integrasi database yang konsisten, berorientasi subject dan *historical data*. *Data warehouse* mengintegrasikan data dari banyak sistem yang tidak kompatibel menjadi suatu bentuk yang menyediakan satu tampilan yang konsisten mengenai perusahaan. Dengan mentransformasikan data menjadi informasi yang berguna, *data warehouse* mengijinkan si pengambil keputusan untuk melakukan analisis lebih sesuai dengan kenyataan , akurat dan konsisten.

2. *Potensi ROI (Return On Investment) yang besar*

   Suatu perusahaan akan mengeluarkan sumber daya yang cukup besar untuk mengimplemtasikan *data warehouse* dan pengeluaran yan berbeda-beda sesuai dengan variasi solusi teknikal yang akan diterapkan pada perusahaan. Bagaimana pun juga. Suatu studi oleh International Data Corporation (IDC)

pada tahun 1996 melaporkan bahwa rata-rata tiga tahun return of investment (ROI) dalam *data warehouse* mencapai 401% dengan lebih dari 90% dari perusahaan yang disurvei mencapai lebih dari 40% ROI, setengah dari perusahaan mencapai lebih dari 160% ROI, dan seperempat lebih mendapat lebih dari 600% ROI (IDC, 1996);

3. *Competitive Advantage*

Return on investment yang besar dari perusahaan yang berhasil mengimplementasikan suatu *data warehouse* adalah bukti dari sangat besarnya *competitive advantage* yang dapat diperoleh dengan menggunakan teknologi in. *Competitive advantage* diperoleh dengan mengijinkan si pengambil keputusan untuk mengakses data tersembunyi yang sebelumnya tidak tersedia, tidak di ketahui, dan tidak dimanfaatkan seperti data mengenai pelanggan, tren, dan permintaan.

Berikut adalah contoh-contoh peluang yang ada karena ketersediaan *informasi strategic* menurut Ponniah (2001):

- ✓ Ketersediaan informasi *strategic* di salah satu bank terbesar di United States dengan asset $250 billion memberikan kesempatan pada users untuk membuat keputusan yang cepat untuk mempertahankan nilai mereka pada *customer*.
- ✓ Pada kasus organisasi pelayanan kesehatan yang besar, terjadi peningkatan yang signifikan program-program pelayanan kesehatan yang terealisasi, dengan hasil 22% penurunan kunjungan *emergency room*, 29% penurunan terhadap pasien anak asma, diabetes dan peningkatan tingkat vaksinasi dan lebih 100.000 *performance report* dibuat untuk pasien dan apoteker.
- ✓ Komunitas apoteker yang bersaing dalam skala nasional dengan lebih dari 800 *franchise* apotik mengerti betul apa yang dibutuhkan oleh *customer*, hasilnya penurunan *level inventory*, meningkatkan efektifitas promosi dan *marketing*, meningkatkan keuntungan bagi perusahaan.

2.1.5 Kategori Data pada *Data Warehouse*

Untuk memahami *data warehouse* lebih dalam, ada dua aspek penting yang harus di pahami yaitu pertama adalah memahami tipe spesifik (*classification*) dari data yang akan disimpan di *data warehouse* dan kedua mengenai tahapan proses dalam pembuatan *data warehouse* . Mengenai kategori pada *data warehouse*, kategori ini diakomodasikan berdasarkan *time-dependent data sources.*

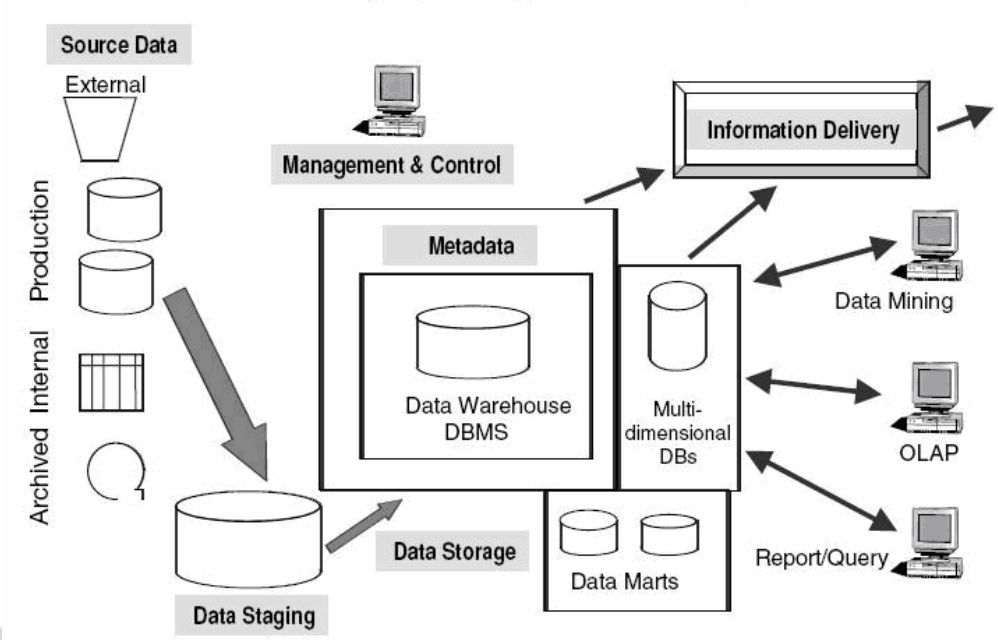
Adapun klasifikasinya adalah sebagai berikut ini: (Kantardzic, 2003)

1. *Old* detail data (data lama)
2. *Current* (new) detail data (data saat ini atau baru)
3. *Lightly summarize* data (data yang disimpulkan secara ringan)
4. *Highly summarize* data (data yang disimpulkan secara berat)
5. *Metadata* (direktori data atau panduan tentang data)

2.1.6 Arsitektur *Data Warehouse*

Arsitektur dalam *data warehouse* mencakup pengaturan yang benar komponen-komponen penyusun *data warehouse,* baik itu *software* ataupun *hardware*. Untuk mempermudah proses pembangunan suatu *data warehouse* diperlukan pemilihan arsitektur yang tepat dan pemahaman yang baik terhadap arsitektur *data warehouse*. Berikut adalah komponen-komponen penyusun *data warehouse.*

Pada sub bab ini ini di jelaskan mengenai arsitektur dan komponen utama dari *data warehouse* (Anahory dan Murray, 1997) beserta proses *tools,* dan teknologi yang berhubungan dengan *data warehouse*. Untuk lebih jelasnya dapat dilihat pada Gambar 2.1 berikut ini.



Gambar 2.1 Arsitektur data *warehouse* (Ponniah, 2001)

- Komponen Sumber Data
    - *Production* Data
    - Internal Data
    - Archived Data
    - External Data
- Data Staging Component
- Data Extraction
- Data Transformation
- Data Loading
- Data Storage Component
- Information Delivery Component
- Metadata Component
- Managament and Control Component

2.1.7 Tahapan *Data Warehouse*

1. Studi kelayakan, pada *phase* ini melakukan kajian *strategic analysis*, termasuk mengevaluasi bisnis line organisasi. Studi kelayakan mencakup mendefinisikan aktifitas-aktifitas, biaya-biaya, keuntungan, faktor-faktor untuk kesuksesan sistem dimasa yang akan datang.
2. Analisa bentuk perusahaan, pengertian/pengetahuan bisnis yang dijalankan dan mengidentifikasi kebutuhan bisnis *(business requirements identification).*
3. Perancangan arsitektur *data warehouse*, arsitektur *logic* dan arsitektur fisik. Tahap ini dilakukan setelah komponen-komponen dalam organisasi didefinisikan terlebih dahulu.
4. Pemilihan teknologi sebagai solusi, mengidentifikasi teknologi-teknologi yang mungkin di gunakan untuk implemetasi arsitektur data dan arsitektur aplikasi dan juga untuk arsitektur sistem *support*.
5. Perencanaan iterasi project, implementasi *data warehouse* dengan satu *subject area* dalam satu waktu yang diukur dengan skala prioritas dan kebutuhan bisnis.
6. Detail *designing* (*data warehouse modeling*), adalah pemodelan *data warehouse* secara lengkap.

Memilih tahapan atau proses yang tepat untuk mengkonstruksi *data warehouse* adalah langkah yang kritis dalam pembuatan suatu *data warehouse.* Data pada *data warehouse* harus distandarisasi terlebih dahulu sebelum dimasukkan. Proses yang digunakan dalam memproses data sebelum dimasukkan ke dalam suatu data *warehouse* adalah proses ETL (*extract, transform and load)*. Penjelasan dari masing-masing adalah sebagai berikut;

1. Extract

Proses *extract* adalah proses mengekstrak (*extracting*) dan pengambilan data dari sumber pada sistem untuk diload ke *data warehouse*. Sumber data dapat

diperoleh secara alternatif melalui ODS (*Operational Data Storage*). Data harus dikonstruksi ulang sebelum dimasukkan ke dalam *data warehouse*.

Proses konstruksi ini melibatkan proses:

a) *Cleansing* data, yaitu proses pembersihan data kotor. Kotor dalam hal ini adalah data berkualitas rendah seperti ketidakkonsistenan penulisan nama, kode id, duplikasi data, data tidak lengkap dan lain-lain.

b) Restrukturisasi data pada *data warehouse* untuk memahami kebutuhan yang ada. Contoh: penambahan atau pengurangan *fields* dan *denormalisasi data*

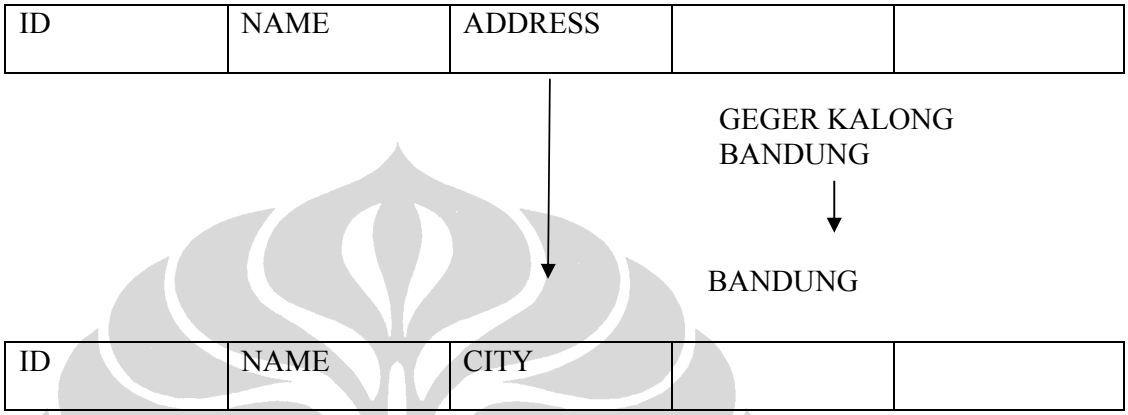
c) Memastikan sumber data konsisten dengan data yang sudah ada di dalam *data warehouse*

2. Transform

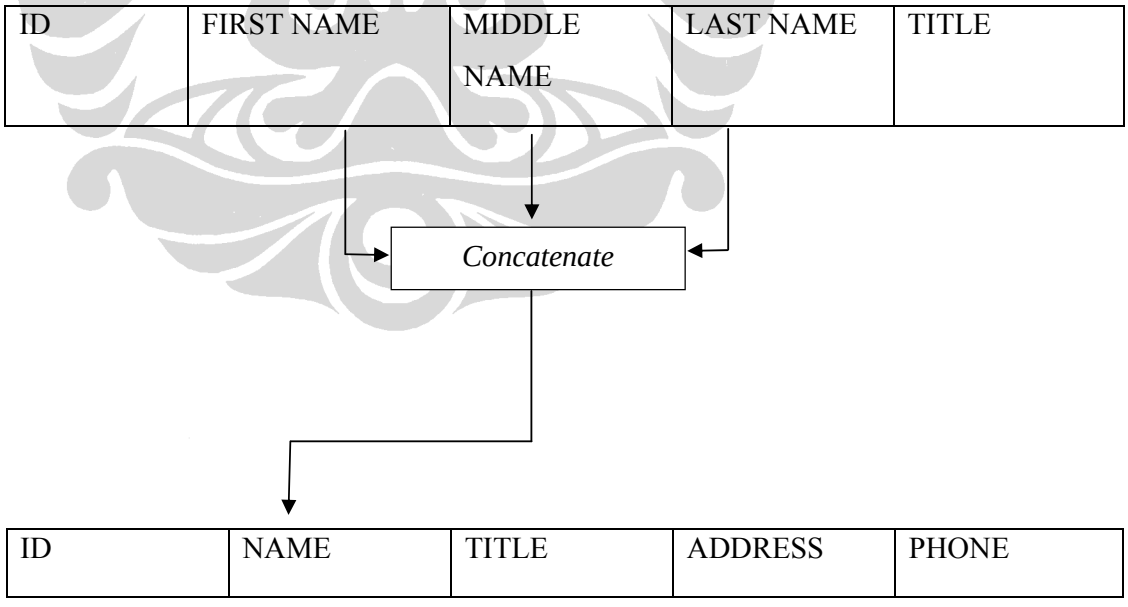
Proses *transform* adalah proses pengubahan data, dimana dat yang diperoleh dari proses extract (dalam format operasional) menjadi data dalam bentuk *data warehouse*. Proses transform ini melibatkan proses:

1. *Summarizing the data*, dengan cara pemilihan (*selection*), proyeksi (*projecting*) , penggabungan (*joining*), normalisasi (*normalization*), agregasi (*aggregation*) dan *grouping* relasional data menjadi *views* yang lebih nyaman dan berguna bagi pengguna.
2. *Packaging the data,* dengan mengkonversi *detail data* atau *summarized data* menjadi format yang lebih berguna seperti *spredsheets, text documents, private database* , dan lainnya.

Terdapat dua cara transformasi yaitu dengan menggunakan fungsi record-level, dan fungsi field-level. Fungsi record-level melibatkan proses *Summarizing the data*, sedangkan fungsi *field-level* adalah *single-field* dan *multi-field.* Transformasi *single-field* mengubah satu *field* menjadi *field* yang lain. Berbeda dengan *multi-field* dimana satu data atau lebih diubah menjadi field baru. Gambar mengenai *single-field* dan *multi-field* dapat dilihat pada Gambar 2.2 dan 2.3.



Gambar 2.2 – Contoh transformasi *single-field* (Zain, 2008)



Gambar 2.3 – Contoh transformasi *multi-field* (Zain, 2008)

3. Load

Proses *load* adalah proses tahapan terakhir dari proses ETL. Pada proses ini akan dilakukan proses pemuatan data dari proses *transform* ke dalam suatu *data warehouse*. Pada proses ini dilakukan juga proses *indexing* untuk memberikan indeks ke masing-masing data untuk mempercepat proses *query*. Terdapat dua mode *loading* ke dalam *data warehouse* yaitu *refresh* dan *update.* Mode *refresh* yaitu proses menuliskan kembali keseluruhan data di dalam *data warehouse* pada suatu interval waktu. Sedangkan untuk mode *update* yaitu suatu proses untuk meng-*update* (tidak menghapus atau menimpa data lain) data yang berubah ke tempat tujuan pada *data warehouse*. Mode *refresh* digunakan pertama kali ketika *data warehouse* berjalan dan hendak dimuat, sedangkan mode *update* umumnya digunakan ketika pemeliharaan data atau ketika *data warehouse* sedang *running*.

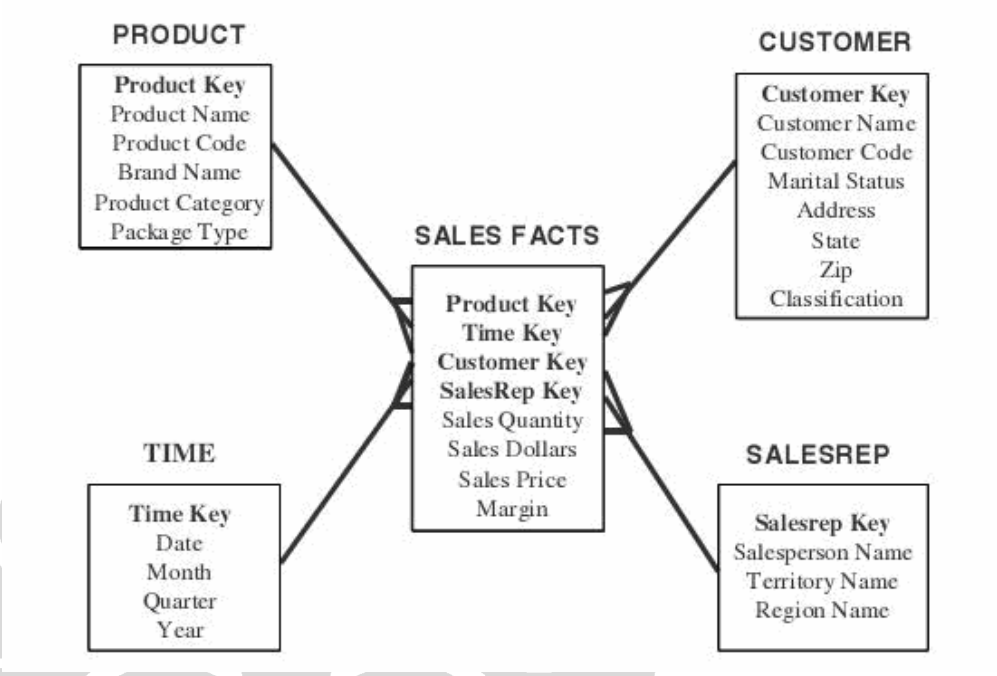
2.1.8 Desain *Data Warehouse*

Untuk memulai pembuatan *data warehouse database* harus memperhatikan keperluan yang utama dan memilih data yang harus didahulukan terlebih dahulu, baru setelah itu bisa diperoleh komponen-komponen *database* dari *data warehouse* adalah *dimensional modelling* (DM). Pengertian dari *dimensional modeling* adalah suatu teknik desain secara logikal yang memiliki sasaran untuk mempresentasikan data dengan standar, bentuk intuitif yang mengijinkan akses secara sangat cepat. Setiap tabel *dimensional model* memiliki komposisi dari satu tabel dengan *composite key* yang dinamakan *fact table* dan sekumpulan set tabel yang lebih kecil yang dinamakan *dimension table.*

Dimensional modelling memiliki beberapa struktur skema, yaitu:

- ✓ Skema bintang (*Star Schema*)

  Struktur logical yang memiliki *fact table* mengandung data fakta posisi tengah, dikelilingi oleh *dimension tables* yang mengandung referensi data (yang bisa *didenormalisasi*). Contoh dapat dilihat pada Gambar 2.4

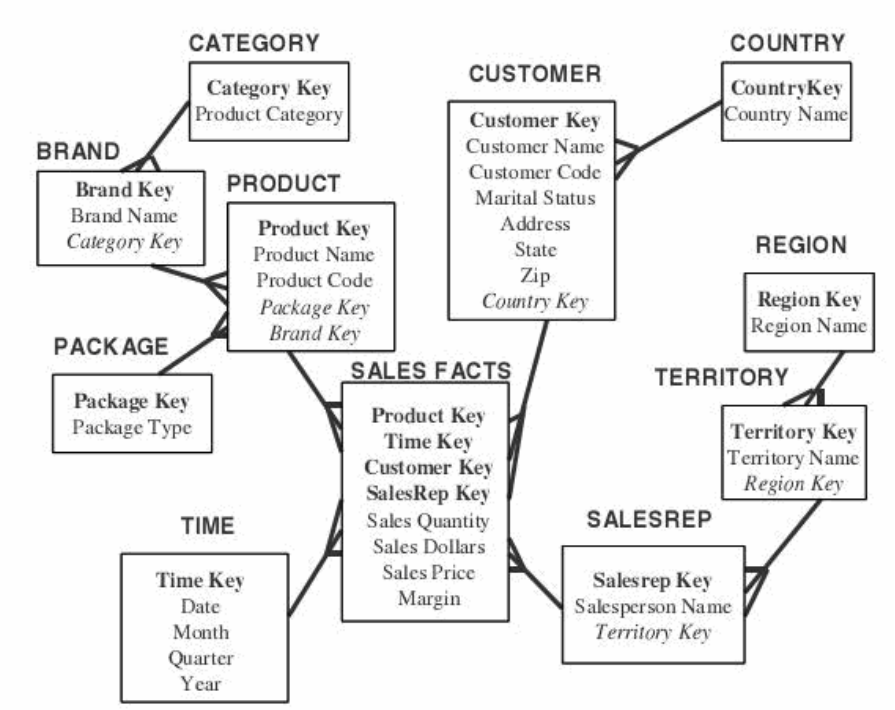


Gambar 2.4 *Star Schema* (Ponniah, 2001)

✓ Skema bola salju (*snowflake schema*)

S*nowflake schema* merupakan perluasan dari skema bintang dengan tambahan beberapa tambahan tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta. Tabel dimensi tersebut berhubungan dengan tabel dimensi yang lain. Skema ini memperbolehkan dimensi memiliki dimensi.

Varian dari *star schema* dimana tabel dimensi tidak mengandung denormalisasi data. Sebagai contoh kita bisa melakukan normalisasi *location data(*atribut *city, region, dan country)* pada *branch* di *dimension table* dari 2.4 untuk membuat *dimension tables* dari *property sales schema* ditampilkan pada Gambar 2.5



Gambar 2.5 *Snow flake Schema* (Ponniah, 2001)

2.2 *Data Mining*

2.2.1 Definisi *Data Mining*

Data mining adalah suatu proses mengekstraksi secara valid, sebelumnya belum diketahui, komprehensif dan informasi yang dapat memberikan aksi dari *database* besar dan menggunakannya untuk membuat keputusan bisnis yang krusial (Simoudis, 1996). *Data mining* berhubungan dengan analisis dari data dan penggunaan teknik *software* untuk menemukan pola yang tersembunyi dan tidak diharapkan dan relasinya dalam bentuk set suatu data. Fokus dari *data mining* adalah memunculkan informasi yang tersembunyi dan tidak diharapkan. Informasi yang tersembunyi tersebut dapat memberikan nilai tambah pada bisnis perusahaan. Selain alasan diatas terdapat pula alasan-alasan lain mengapa diperlukan penggunaan *data mining* berikut ini;

- Data yang tersedia berjumlah sangat besar

  Dalam dekade terakhir ini, harga dari perangkat keras terutama *hardisk* telah turun secara drastis. Disamping itu perusahaan telah mengumpulkan sejumlah data yang sangat besar dari banyak aplikasi yang dimiliki. Dengan sejumlah data ini, perusahaan melakukan eksplorasi data untuk mencari pola tersembunyi sebagai panduan untuk membantu strategi bisnis yang akan dijalankan
- Kompetisi yang meningkat

  Kompetisi yang ada sangat tinggi sebagai hasil dari *marketing* dan dengan adanya saluran distribusi seperti internet dan telekomunikasi. Perusahaan akan menghadapi kompetisi dunia, karena itu kunci suksesnya bisnis adalah kemampuan untuk membina pelanggan yang sudah ada dan mendapatkan yang baru. Teknologi *data mining* dapat membantu perusahaan untuk menganalisa faktor yang mempengaruhi hal tersebut.
- Kemampuan Teknologi

  Teknologi *data mining* sebelumnya hanya ada pada lingkungan akademik tetapi saat ini banyak teknologi seperti ini semakin canggih dan siap untuk diterapkan pada industri. Algoritma yang ada semakin akurat, efisien dan dapat menangani komplikasi data yang meningkat. Sebagai tambahan, *data mining application programming interfaces* (*APIs*) telah distandarisasi, sehingga mengijinkan pengembang untuk membuat aplikasi *data mining* yang lebih baik.

2.2.2 Teknik *Data Mining*

Ada empat operasi utama yang dapat dilakukan pada teknik *data mining* yaitu *predictive modelling, database segmentation, link analysis*, dan *deviation detection.* Meskipun salah satu dari operasi utama dapat digunakan untuk mengimplementasikan aplikasi bisnis apapun, ada keterhubungan yang ditemukan antara aplikasi dan operasi yang bersangkutan. Teknik adalah implementasi secara spesifik dari operasi *data mining*. Bagaimanapun juga masing-masing operasi memiliki kekuatan dan kelemahannya masing-masing. Untuk lebih jelasnya

mengenai teknik yang berasosiasi dengan salah satu dari empat operasi utama *data mining* (Cabena 1997 dalam Zein 2008) dapat dilihat pada Tabel 2.1 berikut ini:

Operations	**Data Mining *Techniques***
Predictive modelling	*Classification* *Value Prediction*
Database segmentation	*Demographic clustering* *Neural clustering*
Link analysis	*Association discovery* *Sequential pattern discovery* *Similar time sequence discovery*
Deviation detection	*Statistics* *Visualization*

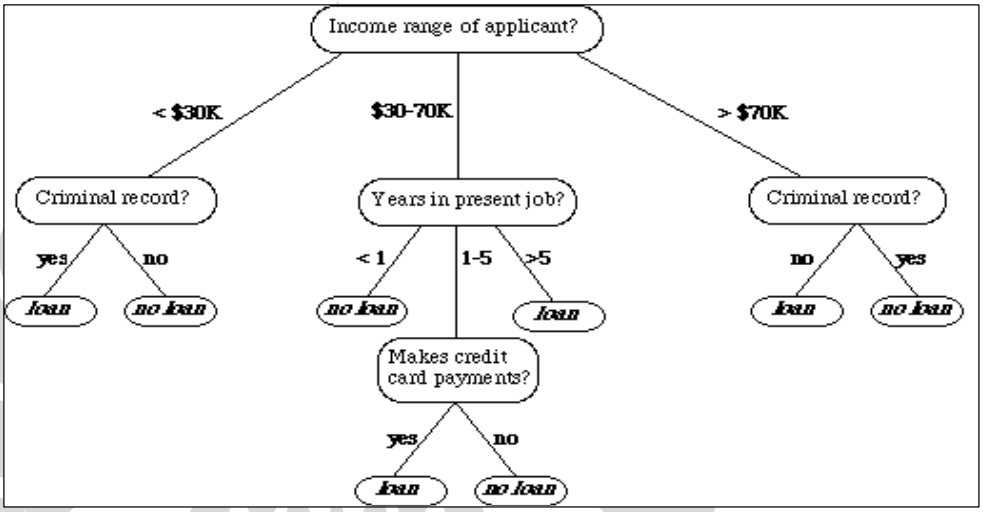
Tabel 2.1 Tabel operasi *data mining* dan teknik yang digunakannya (Zein, 2008)

2.2.2.1 Predictive Modeling

Predictive Modeling menggunakan pendekatan generalisasi dari '*real world*' dan kemampuan menempatkan data baru ke kerangka utama. *Predictive modeling* bisa digunakan untuk menentukan karakteristik (model) mengenai data set. Model ini dikembangkan dengan menggunakan *supervised learning* yang terdiri dari dua fase: *training dan testing. Training* membuat model menggunakan sampel besar dari data yang dinamakan training set, sedangkan *testing* mencoba model baru, data yang sebelumnya tidak terlihat untuk menentukan keakurasian dan karakteristik performa fisik. Ada dua teknik yang berasosiasi dengan *predictive modeling: classification* dan *value prediction.*

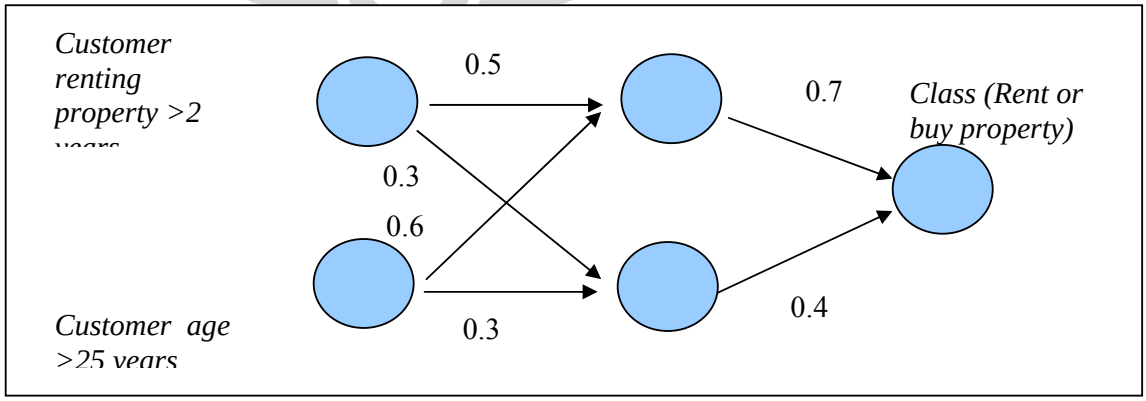
Classification digunakan untuk membangun kelas spesifik yang telah ditentukan sebelumnya untuk masing-masing record di *database,* dari suatu set terbatas ke nilai kelas yang memungkinkan. Ini adalah dua spesialisasi dari klasifikasi: *tree induction dan neural induction.* Contoh dari klasifikasi menggunakan *induction* ada pada Gambar 2.7. Pada contoh ini menggambarkan

bagaimana institusi keuangan memuruskan kelayakan calon nasabahnya layak diberikan pinjaman atau tidak. *Predictive model* telah menentukan bahwa dua variabel yang digunakan yaitu: range pendapatan applicant. catatan kriminal dan lama waktu selama bekerja. Model ini membantu menentukan *applicant* yang layak mendapatkan pinjaman, jika *applicant* memliki pendapatan antara $30-$70 dan bekerja kurang dari setahun ditempat kerja terkahir, maka *applicant* ini tidak boleh diberikan pinjaman.



Gambar 2.6 *Clasification* menggunakan *tree induction*
(http://www.cse.unsw.edu.au/~billw/cs9414/notes/ml/06prop/id3/id3.html)

Contoh *Classification* dengan mengunakan *neural induction* ditunjukkan pada Gambar 2.7



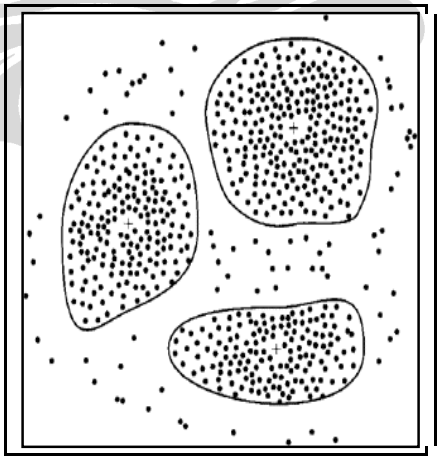
Gambar 2.7 *Clasification* menggunakan *neural network* (Connonly and Begg, 2005)

Pada kasus ini, *classification* dari data diperoleh dengan menggunakan *nueral network*. *Neural network* mengadung koleksi dari titik-titik yang terkoneksi dengan input, output, dan processing pada masing-masing titik. Antara lapisan input dan output mungkin sebagai sejumlah lapisan proses tersembunyi. Masing-masing proses unit dalam satu lapisan saling berhubungan dengan proses unit di lapisan berikutnya oleh *weighted value* yang menggambarkan kekuatan hubungan.

2.2.2.2 *Database Segmentation* dengan *Clustering*

Berbeda dengan *clasification* dimana kelas data telah ditentukan sebelumnya, *clustering* melakukan pengelompokan data tanpa berdasarkan kelas data tertentu. Bahkan *clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui itu. Karena itu *clustering* sering digolongkan sebagai metode *unsupervised learning*.

Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas/*cluster*. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang multidimensi. Ilustrasi dari *clustering* dapat dilihat di Gambar 2.8 dimana lokasi, dinyatakan dengan bidang dua dimensi, dari pelanggan suatu toko dapat dikelompokkan menjadi beberapa *cluster* dengan pusat *cluster* ditunjukkan oleh tanda positif (+).



Gambar 2.8 Contoh *Clustering* (Ponniah, 2001)

Beberapa kategori algoritma *clustering* yang banyak dikenal adalah metode partisi dimana pemakai harus menentukan jumlah k partisi yang diinginkan lalu setiap data dites untuk dimasukkan pada salah satu partisi, metode lain yang telah lama dikenal adalah metode hierarki yang terbagi dua lagi : *bottom-up* yang menggabungkan cluster kecil menjadi *cluster* lebih besar dan *top-down* yang memecah *cluster* besar menjadi *cluster* yang lebih kecil. Kelemahan metode ini adalah bila salah satu penggabungan/pemecahan dilakukan pada tempat yang salah, tidak dapat didapatkan *cluster* yang optimal. Pendekatan yang banyak diambil adalah menggabungkan metode *hierarki* dengan metode *clustering* .

2.2.2.3 Link Analysis

Link analysis memiliki sasaran untuk membangun jaringan yang dinamakan *associations* antara *individual records atau sets of records* di dalam *database.* Ada tiga spesialisasi dari analysis jaringan: *associations discovery, sequential pattern discovery* dan *similar time sequence discovery. Associations discovery* digunakan untuk menemukan item yang menyatakan keberadaan dari item yang lain didalam *event* yang sama. *Sequencial pattern discovery* menemukan *pattern* antara *event* seperti keberadaan satu set dari kelompok item yang diikuti oleh satu set dari sekelompok *item* yang diikuti oleh satu set dari sekelompok *item* didalam *database* dalam beberapa periode waktu. *Similar time sequence discovery* digunakan seperti contoh: dalam *discovery of links* antara dua set data yang bergantung terhadap waktu dan berdasarkan derajat kesamaan antara pola dari suatu seri waktu.

Association Rule Mining merupakan bagian dari *Frequent Pattern Mining. Frequent Pattern Mining* merupakan salah satu task *data mining* yang sangat penting. Task ini mencari hubungan/relasi, assosiasi, dan korelasi dalam data. Pengetahuan yang dihasilkan juga sangat berguna untuk klasifikasi, clustering, dan task data mining yang lain. Selain *Association Rule Mining*, masih ada *Sequential Pattern*, dan *Structured Pattern* yang termasuk dalam *Frequent Pattern Mining. Association Rule Mining* dapat juga disebut *Frequent Itemset Mining* karena pola yang dihasilkan adalah pola item yang sering muncul bersamaan dalam sebuah *database*. Contoh klasik yang sering digunakan untuk

menjelaskan *Association Rule Mining* adalah market basket analisis. Pada market basket analisis, kita menganalisa kebiasaan konsumen dalam membeli barang.

Secara umum, *Association Rule Mining* dapat dibagi menjadi dua tahap yaitu pencarian Frequent Itemset *(Frequent Itemset Candidate Generation)* dan Rule Generation. Pada tahap *Frequent Itemset Candidate Generation* terdapat beberapa kendala yang harus dihadapi untuk memperoleh *Frequent Itemset* seperti banyaknya jumlah kandidat yang memenuhi minimum support, dan proses perhitungan minimum support dari Frequent Itemset yang harus melakukan scan database berulang-ulang. Pendekatan apriori sangat membantu dalam mengurangi jumlah kandidat *Frequent Itemset*.

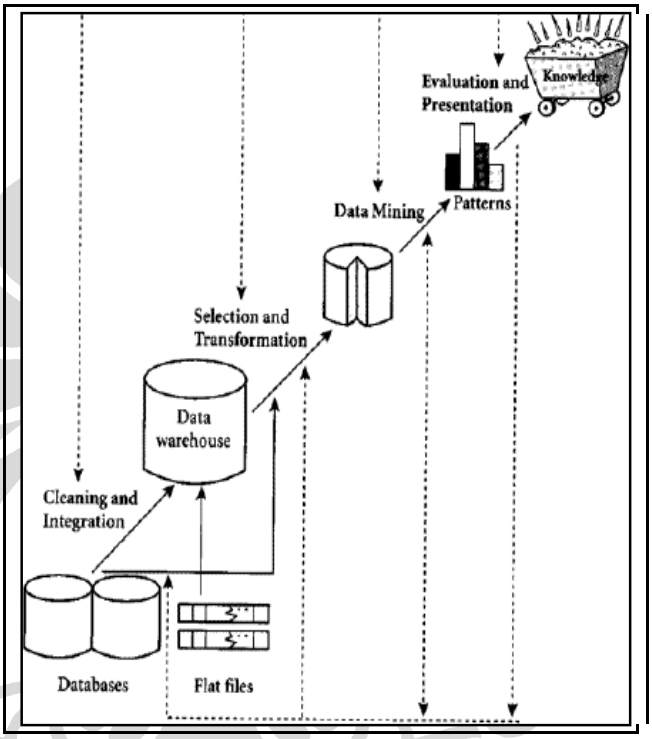
Dengan menggunakan *FP-growth*, dapat dilakukan *Frequent Itemset Mining* tanpa melakukan *candidate generation*. *FP-growth* menggunakan struktur data FP-tree. Dengan menggunakan cara ini scan database hanya dilakukan dua kali saja, tidak perlu berulang-ulang. Data akan direpresentasikan dalam bentuk *FP-tree*. Setelah *FP-tree* terbentuk, digunakan pendekatan divide and conquer untuk memperoleh *Frequent Itemset. FP-tree* merupakan struktur data yang baik sekali untuk *Frequent Pattern mining*. Struktur ini memberikan informasi yang lengkap untuk membentuk *Frequent Pattern*. Item-item yang tidak frequent (infrequent) sudah tidak ada dalam *FP-tree*.

2.2.2.4 *Deviation Detection*

Deviation detection adalah teknik yang relatif masih baru dalam teknik *data mining*. Namun *Deviation detection* sering kali menjadi sumber penemuan baru karena teknik ini mengidentifikasi outlier yang mengekspresikan deviasi dari penemuan sebelumnya. Operasi ditampilkan menggunakan teknik *statistics* dan *visulazation* atau sebagai suatu produk dari data mining. sebagai contoh regresi linier memfasilitasi pengidentifikasian data dalam teknik visualisasi modern yang menampilkan kesimpulan dan representasi grafik yang membuat deviasi mudah untuk dideteksi.

2.2.3 Tahap-Tahap *Data Mining*

Sebagai suatu rangkaian proses, *data mining* dapat dibagi menjadi beberapa tahap yang diilustrasikan di Gambar 2.9. Tahap-tahap tersebut, bersifat interaktif dimana pemakai terlibat langsung atau dengan perantaraan *knowledge base.*



Gambar 2.9 Tahap-tahap *data mining* (Zein, 2008)

- *Pembersihan data (untuk membuang data yang tidak konsisten dan noise)*

  Pada umumnya data yang diperoleh, baik dari database suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa data mining yang kita miliki. Data-data yang tidak relevan itu juga lebih baik dibuang karena keberadaannya bisa mengurangi mutu atau akurasi dari hasil data mining nantinya. ”*Garbage in garbage out”* (hanya sampah yang akan dihasilkan bila yang dimasukkan juga

sampah) merupakan istilah yang sering dipakai untuk menggambarkan tahap ini. Pembersihan data juga akan mempengaruhi performa dari sistem *data mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

- *Integrasi data (penggabungan data dari beberapa sumber)*

  Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu database tetapi juga berasal dari beberapa database atau file teks. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dsb. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada. Dalam integrasi data ini juga perlu dilakukan transformasi dan pembersihan data karena seringkali data dari dua database berbeda tidak sama cara penulisannya atau bahkan data yang ada di satu database ternyata tidak ada di database lainnya.

  Hasil integrasi data sering diwujudkan dalam sebuah *data warehouse* karena dengan *data warehouse*, data dikonsolidasikan dengan struktur khusus yang efisien. Selain itu *data warehouse* juga memungkinkan tipe analisa seperti OLAP. Untuk membangun *data warehouse* juga tersedia paket-paket *software* yang mapan seperti database-nya dan piranti pendukung yang sering disebut sebagai ETL (*Extract Transform Loading*). Banyak paket *software* ETL sudah mencakup tahap pembersihan dan integrasi data.

- *Transformasi data (data diubah menjadi bentuk yang sesuai untuk di-mining)*

  Beberapa teknik *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa teknik standar seperti analisis *asosiasi* dan *clustering* hanya bisa menerima input data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi

menjadi beberapa interval. Proses ini sering disebut *binning*. Disini juga dilakukan pemilihan data yang diperlukan oleh teknik *data mining* yang dipakai. Transformasi dan pemilihan data ini juga menentukan kualitas dari hasil data mining nantinya karena ada beberapa karakteristik dari teknik-teknik data mining tertentu yang tergantung pada tahapan ini.

- *Aplikasi teknik data mining*

  Aplikasi teknik *data mining* sendiri hanya merupakan salah satu bagian dari proses *data mining*. Ada beberapa teknik data mining yang sudah umum dipakai. Kita akan membahas lebih jauh mengenai teknik-teknik yang ada di seksi berikutnya. Perlu diperhatikan bahwa ada kalanya teknik-teknik *data mining* umum yang tersedia di pasar tidak mencukupi untuk melaksanakan *data mining* di bidang tertentu atau untuk data tertentu. Sebagai contoh akhir-akhir ini dikembangkan berbagai teknik *data mining* baru untuk penerapan dibidang bioinformatika seperti analisa hasil *microarray* untuk mengidentifikasi DNA dan fungsi-fungsinya.

- *Evaluasi pola yang ditemukan (untuk menemukan yang menarik/bernilai)*

  Dalam tahap ini hasil dari teknik *data mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa ada beberapa alternatif yang dapat diambil seperti : menjadikannya umpan balik untuk memperbaiki proses *data mining*, mencoba teknik *data mining* lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

  Ada beberapa teknik *data mining* yang menghasilkan hasil analisa berjumlah besar seperti analisis asosiasi. Visualisasi hasil analisa akan sangat membantu untuk memudahkan pemahaman dari hasil *data mining.*

- *Presentasi pola yang ditemukan untuk menghasilkan aksi*

  Tahap terakhir dari proses *data mining* adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisa yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami data

mining. Karenanya presentasi hasil data mining dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses data mining. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil *data mining*.

2.3 Bisnis *Airline* dan *Data Mining*

Pelayanan terhadap penumpang yang bersifat individu sebagai hasil dari segmentasi terhadap pelanggan harus dapat dijelaskan dan nilai tambahnya juga harus dapat dibuktikan. Hasil segmentasi terhadap penumpang adalah mendapatkan pemahaman yang lebih baik mengenai konsumen *airline*. Pritscher (2001) mengatakan, peluang *data mining* dalam industri *airline* dapat dilakukan dengan memilah-milah *customer* yang katagori pembagian tersebut dapat dilakukan berdasarkan:

1 *Region* : Kebutuhan penumpang akan terlayani dengan lebih baik jika diketahui geografi yang disukai diketahui dengan baik atau tempat penumpang. Penumpang dapat dipilah-pilah berdasarkan asal dan tujuan perjalanan atau branch office yang terdekat dengan tempat tinggal penumpang.

2 *Market*: Dari sudut pandang sales, market dibedakan menjadi *market home*, *market* pihak ketiga yang dilayani karena penumpang dari airline lain (*connecting passenger*). Penumpang dapat dipilah berdasarkan *market* yang mereka gunakan dan tempat tinggal mereka.

3 *Travel Preference*: Agar dapat melayani penumpang dengan penawaran yang tepat, diperlukan perjananan/travel yan bagaimana yang diinginkan oleh penumpang. Penumpang dapat bedakan berdasarkan rasio dari penerbangan *longhaul* (*intercontinental*) dan penerbangan *shorthaul*, berdasarkan *rasio* dari tiket yang *hight fare* dan *low fare* juga kombinasi keduanya.

4 *Travel Behaviour*: Salah satu informasi penting pelanggan adalah jenis tiket yang dibeli secara berkala. Tipe tiket ini terkait erat dengan tujuan

perjalanan misalnya untuk berlibur atau untuk perjalanan bisnis. Data penting yang tercatat adalah *booking class* yang digunakan untuk penerbangan, sedangkan waktu dan harga tiket tidak terekam karena tergantung pada ketersediaan dan waktu.

Menurut McIlroy dan Barnett (2000) dalam Wijaya (2005), biaya untuk mendapatkan konsumen yang baru dapat mencapai 5 kali dari mempertahankan konsumen yang sudah ada. Keanggotaan dalam *loyalty program* dibagi menjadi 2 jenis, *open* dan *limited*. *Limited loyalty* tidak dapat dikuti oleh semua orang, ada mekanisme tertentu untuk menjadi anggota *limited* ini seperti membayar uang pendaftaran dan kadang-kadang disertai syarat-syarat lainnya seperti melakukan pmbelian dengan volume tertentu atau memiliki penghasilan tertentu (Wijaya, 2005).

2.3.1 *Loyalty Program* dan *Frequent Flyer Program*

Menarik dan mempertahanan konsumen memerlukan biaya yang tinggi, khususnya untuk industri jasa. *Loyalitas* didefinisikan Oliver (Celuch dan Goodwin, 1999) sebagai komitmen yang tinggi untuk membeli kembali suatu produk atau jasa yang disukai di masa mendatang, disamping pengaruh situasi dan usaha pemasar dalam merubah perilaku. Dengan kata lain konsumen akan setia untuk melakukan pembelian ulang secara terus-menerus.

Menurut Shoemaker dan Lewis (1998) dalam Wijaya (2005), *loyalty program* adalah adalah program yang ditawarkan pada konsumen untuk membangun ikatan terhadap merek/*brand* tertentu. Lebih lanjut Wijaya (2005) menyatakan bahwa sebagian besar konsumen melakukan pembelian ulang dalam rangka menambah keuntungan yang ditawaran kemudian *redeem* dengan menggunakan *reward* yang telah dikumpulkan. Konsumen loyal terhadap program, bukan kepada perusahaannya. Tidak ada hubungan langsung antara program dengan ikatan emosional konsumen terhadap perusahaan. Hubungan emosional dapat terbentuk salah satunya dengan memberikan pelayanan yang baik.

Lebih dalam lagi Gramer dan Brown (Utomo, 2006) memberikan definisi mengenai *loyalitas* (loyalitas jasa), yaitu derajat sejauh mana seorang konsumen menunjukkan perilaku pembelian berulang dari suatu penyedia jasa, memiliki suatu desposisi atau kecenderungan sikap positif terhadap penyedia jasa, dan hanya mempertimbangkan untuk menggunakan penyedia jasa ini pada saat muncul kebutuhan untuk memakai jasa ini. Dari definisi yang disampaikan Gramer dan Brown, konsumen yang *loyal* tidak hanya seorang pembeli yang melakukan pembelian berulang, tetapi juga mempertahankan sikap positif terhadap penyedia jasa.

Frequent flyer program (FFP) adalah *loyalty program* yang ditujukan bagi para pengguna jasa penerbangan. *Airline* pengelola FFP memberikan poin reward pada konsumen (disebut *miles*) yang bisa digunakan untuk membeli tiket (Emch, 2007). Ada 3 alasan bagaimana FFPs dapat mengurangi ongkos produksi dalam dunia penerbangan (Adrian Emch , 2007);

1. Mempertahankan konsumen yang sudah jadi pelanggan lebih murah dibandingkan dengan mencari pelanggan baru.
2. Pelanggan yang setia lebih menguntungkan dari pada pelangggan baru
3. FFP dapat digunakan untuk meningkatkan pelayanan melalui personalisasi dari setiap *service* yang diberikan untuk penumpang.

Pada prinsipnya setiap tiket gratis yang dibeli dengan menggunakan *mileage* yang dimiliki oleh anggota FFP adalah tiket untuk tempat duduk yang kosong. Tidak ada biaya yang harus dikeluarkan untuk tiket yang dikeluarkan, hanya pada tataran praktis, hal ini belum dikaji mekanismenya. Di GFF sendiri, setiap tiket yang dibeli oleh penumpang ada prosentase bagian *awardnya*.

2.4 *Tools Development*

2.4.1 Pentaho Data Integration (PDI/Kettle)

Kettle adalah aplikasi ETL (*Extract, Transformation and Load*). Aplikasi Kettle sendiri merupakan bagian dari aplikasi BI Pentaho. Sebelumnya proyek ini berdiri sendiri dan kemudian diakuisisi oleh Pentaho pada tahun 2006.

Kettle terdiri dari 4 aplikasi, yaitu :

- ✓ *Spoon*, yaitu aplikasi grafis berbasis swing yang digunakan untuk merancang file skema *job* dan *transformation*
- ✓ *Pan*, yaitu *script* yang digunakan untuk menjalankan file skema *transformation* melalui terminal / *command line*
- ✓ *Kitchen*, yaitu *script* yang digunakan untuk menjalankan file skema ***job*** melalui terminal / *command line*
- ✓ *Carte*, yaitu *temporary web server* yang digunakan untuk mengeksekusi *job/transformation* secara *cluster* atau *parallel*

Kesemua aplikasi tersebut di atas dijalankan melalui *shell* atau *batch script* yang berkaitan. Fitur-fitur Kettle antara lain :

- Memiliki utilitas grafik yang dapat digunakan merancang skema *step* atau langkah kontrol dan transformasi data.
- *Multi platform* - karena dikembangkan di atas Java yang notabene berjalan di banyak *platform*.
- Bersifat *concurrent*, dalam arti *row-row* data diambil oleh suatu step dan diserahkan ke *step* lain secara *parallel*. Artinya tidak menunggu sampai suatu koleksi data diambil secara keseluruhan terlebih dahulu.
- *Scalable* - dapat beradaptasi dengan penambahan kapasitas memori RAM atau pun *storage* (*scale up*) dan dapat beradaptasi dengan penambahan *node* komputer atau cluster lain (*scale out*).

2.4.2 Mondrian

Mondrian adalah *web* aplikasi yang berbasis *open source.* Mondrian memiliki fitur-fitur *untuk* melakukan *drill-down, drill-up* seperti halnya *tools* untuk presentasi *data warehouse* lainnya.

2.5 Kebutuhan *Data Mining* dalam *Frequent Flyer Program*

Frequent Flyer Program menyimpan data-data anggota yang juga pelanggan setia dari penerbangannya. Data penerbangan anggota *Frequent Flyer* adalah sekedar tumpukan data operasional bagi *airline* yang memiliki sistem tersebut. Infrastruktur IT untuk operasional *Frequent Flyer* dapat mudah ditiru oleh *airline* lainnya dan saat itu *Frequent Flyer* bukan lagi menjadi pembeda bagi sebuah *airline* dari pesaingnya karena *Frequent Flyer Program* sudah menjadi komoditas yang mudah dimiliki oleh setiap industri penerbangan.

Penggunaan data yang dimiliki secara ekstensif dapat menjadi keunggulan kompetitif bagi suatu *airline*. Hal ini menjadikan *airline* tersebut berbeda dengan *airline* lainnya ketika data tersebut diolah dan dianalisa dengan metode statistik atau teknik data *mining* lainnya dalam BI. Hasil dari analisa tersebut sulit ditiru oleh perusahaan lain dan bisa menjadi keunggulan kompetitif bagi perusahaan yang bersangkutan.

Sebagai gambaran adalah promo-promo yang diadakan oleh sebuah *airline* bisa menggunakan data anggota FFP agar promo tersebut bisa tepat sasaran. *Airline* juga bisa membuat jadwal penerbangan dan memprediksi penerbangan dari mana kemana yang pada waktu tertentu seperti *peak season* dapat lebih optimal. Juga penggunaan data FFP untuk memberikan *personalisasi* untuk setiap service yang diberikan *airline*.