



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Kemacetan

Kemacetan merupakan sebuah kondisi tersendatnya atau terhentinya lalu lintas yang disebabkan oleh banyaknya jumlah kendaraan melebihi kapasitas jalan. Terjadinya kemacetan diawali dengan mendekatnya *volume* kendaraan melintas terhadap kapasitas jalan yang dilalui. Pendekatan tersebut menurunkan jarak antar kendaraan yang melalui yang mengakibatkan turunnya kecepatan kendaraan. Turunnya kecepatan bebas ruas jalan yang mendekati 0 km/jam mengakibatkan terjadinya antrian. Kemacetan dapat meningkat seiring dengan meningkatnya mobilitas manusia yang menggunakan transportasi, terutama pada saat-saat sibuk (Sudrajat:2011). Kemacetan total terjadi ketika kendaraan harus berhenti atau bergerak sangat lambat (Tamin:2000). Lalu lintas memiliki ketergantungan terhadap kapasitas jalan. Jika lalu lintas yang bergerak melampaui kapasitas jalan, maka lalu lintas akan terhambat dan akan mengalir sesuai dengan kapasitas maksimum jaringan jalan (Sinulingga:1999).

2.1.1 Penyebab Kemacetan

Terdapat tujuh penyebab kemacetan (Administration:2005), yaitu:

- 1 *Physical Bottlenecks*: Kemacetan yang disebabkan oleh jumlah kendaraan yang melebihi batas atau berada pada tingkat tertinggi. Kapasitas tersebut ditentukan dari faktor jalan, persimpangan jalan, dan tata letak jalan.
- 2 Kecelakaan Lalu-lintas (*traffic incident*): Kemacetan yang disebabkan oleh adanya kejadian atau kecelakaan dalam jalur perjalanan.
- 3 Area Pekerjaan (*work zone*): Kemacetan yang disebabkan oleh adanya aktivitas konstruksi pada jalan. Aktivitas tersebut akan mengakibatkan perubahan kondisi lingkungan jalan.
- 4 Cuaca yang Buruk (*bad weather*): Keadaan cuaca dapat menyebabkan perubahan perilaku pengemudi sehingga dapat mempengaruhi arus lalu lintas.
- 5 Alat Pengatur Lalu lintas (*poor signal timing*): Kemacetan yang disebabkan oleh pengaturan lalu-lintas yang bersifat kaku dan tidak mengikuti tinggi rendahnya arus lalu lintas.
- 6 Acara Khusus (*special event*): Merupakan kasus khusus dimana terjadi peningkatan arus yang disebabkan oleh adanya acara-acara tertentu.
- 7 Fluktuasi pada Arus Normal (*fluctuations in normal traffic*): Kemacetan yang disebabkan oleh naiknya arus kendaraan pada jalan dan waktu tertentu.

2.2 *GoogleMaps*

GoogleMaps adalah sebuah layanan peta *online* yang dapat digunakan secara mudah dan gratis oleh pengguna. *GoogleMaps* juga menawarkan *API* (*Application Programming Interface*) yang dapat digunakan oleh *developer* untuk memanfaatkan layanan pada *GoogleMaps* dan menerapkannya pada aplikasi yang dibuatnya. *GoogleMaps API* merupakan suatu *library* dalam bentuk *JavaScript*

Pada *GoogleMaps API* terdapat 4 jenis pilihan *model* peta disediakan oleh Google, diantaranya:

1. *ROADMAP*, untuk menampilkan peta biasa 2 dimensi.
2. *SATELLITE*, untuk menampilkan foto satelit.
3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
4. *HYBRID*, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota).

2.2.1 *Location Based-Service (LBS)*

Location Based Service adalah layanan yang berfungsi untuk mencari lokasi dengan menggunakan teknologi *Global Positioning Service (GPS)* dan *Google's cell location*. Map dan layanan berbasis lokasi menggunakan lintang dan bujur untuk menentukan lokasi geografis.

Dua unsur utama *LBS* adalah:

1. *Location Manager (API Maps)*

Menyediakan *tools/resource* untuk menampilkan, memanipulasi *maps/peta* dan fitur-fitur lainnya seperti tampilan satelit, *street*, maupun gabungannya. Paket ini berada pada *com.google.android.maps*.

2. *Location Providers (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device/perangkat*. *API Location* berhubungan dengan *data GPS* dan *data lokasi realtime*. *API Location* berada pada paket *Android* yaitu dalam paket *android.location*.

2.3 **Web**

Web adalah sebuah kumpulan halaman yang diawali dengan halaman muka yang berisikan informasi, iklan, serta *program* aplikasi (Asropudin:2013). *Web* adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui *internet*) (Ardhana:2012).

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *web* merupakan sebuah kumpulan halaman yang berisikan informasi, iklan, atau program aplikasi yang memiliki konsep *hyperlink* sehingga dapat memudahkan *user* untuk mengakses halaman tersebut.

2.4 *PHP*

PHP atau yang memiliki kepanjangan *PHP Hypertext Processor* merupakan bahasa pemrograman yang memiliki fungsi untuk membangun sebuah *website* yang dinamis (Saputra:2011). *Website* yang dinamis adalah halaman pada *website* yang akan ditampilkan dibuat saat halaman tersebut diminta oleh *client*. Dalam penggunaannya, *PHP* dapat diintegrasikan dengan *HTML*. *HTML* digunakan sebagai kerangka *layout web* sedangkan *PHP* digunakan untuk mengatur proses yang terjadi pada *website*.

PHP merupakan bahasa pemrograman berbasis *server-side* yang dapat melakukan *parsing script* menjadi *script web* sehingga pada sisi *client* menghasilkan tampilan yang menarik (Ardhana:2012).

2.5 *Data*

Data adalah satu *file* ataupun *field* yang berupa karakter dan gambar (Wahyudi:2012). *Data* adalah fakta dari suatu pernyataan yang berasal dari kenyataan, dimana pernyataan tersebut merupakan hasil pengukuran atau pengamatan. *Data* adalah fakta yang sudah ditulis dalam bentuk catatan atau direkam ke dalam bentuk media (Amsyah:1987).

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *data* merupakan sebuah *file* yang tersusun atas karakter maupun gambar. Karakter maupun gambar tersebut didapatkan berdasarkan hasil pengukuran atau pengamatan yang dicatat maupun direkam ke dalam bentuk media.

2.6 Pengolahan *Data*

Pengolahan *data* adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk *data* menjadi informasi yang memiliki kegunaan (Ladjamudin:2013). Pengolahan *data* adalah proses perhitungan atau transformasi *data* input menjadi informasi yang mudah dimengerti maupun sesuai dengan yang diinginkan.

Berdasarkan penjelasan di atas dapat disimpulkan bahwa pengolahan *data* adalah proses perubahan atau transformasi *data* yang melibatkan masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk *data* sehingga menghasilkan informasi yang mudah dimengerti atau sesuai dengan kebutuhan.

2.7 *Database*

Database adalah sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan *data* dan hubungan diantaranya (Sutarman:2012). *Database* adalah sekumpulan *data store* (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*. atau media penyimpanan sekunder lainnya (Ladjamudin:2013).

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *database* merupakan sekumpulan *file* yang saling terhubung dan terorganisir yang menyimpan *data* dan hubungannya. Penyimpanan dapat dilakukan dengan menggunakan media penyimpanan sekunder.

2.8 *Online Analytical Processing*

Online analytical processing (OLAP) adalah sebuah perangkat yang menggambarkan teknologi menggunakan visualisasi multidimensi sejumlah *data* untuk menyediakan akses yang lebih cepat bagi strategi informasi dengan tujuan mempercepat analisis. Dalam model *data OLAP*, informasi di gambarkan secara konseptual seperti kubus (*cube*), yang terdiri atas kategori deskriptif (*dimensions*) dan nilai kuantitatif (*measures*) (Reddy : 2010).

Data multidimensi memiliki *atribut* tersendiri untuk bisa dikelola dalam *OLAP*. Terdapat tiga atribut diantaranya adalah (Feri : 2011):

- a. Dimensi (*dimension*): adalah suatu *atribut* yang di tinjau.
- b. Pengukur (*measurement*): besaran yang dapat diukur mengacu pada irisan antara dimensi yang di tinjau.
- c. Kalkulasi (hasil pengukuran): adalah nilai dari *measurement*.

Pada model data kubikal virtual ada beberapa jenis metode pengoperasian diantaranya mencakup (Feri : 2011):

1. Membuat irisan / *Slicing* adalah mengambil atau mengiris satu dimensi dari *data* virtual kubikal untuk keperluan penyederhanaan informasi atau untuk membuang informasi yang tidak diperlukan dalam analisa.
2. Membuat banyak irisan / *Dicing* adalah irisan yang dilakukan lebih dari dua dimensi data.
3. *Drill Up* dan *Drill Down* merupakan teknik analisa untuk menggeneralisasi atau menspesifikasi informasi, semakin ke atas maka

informasi makin ringkas dan semakin ke bawah maka informasi semakin rinci.

4. Rotasi atau *Pivoting* dilakukan dengan memutar atau merotasi *data* kubikal *virtual* untuk mendapat sudut pandang berbeda terhadap *data* yang dianalisa.

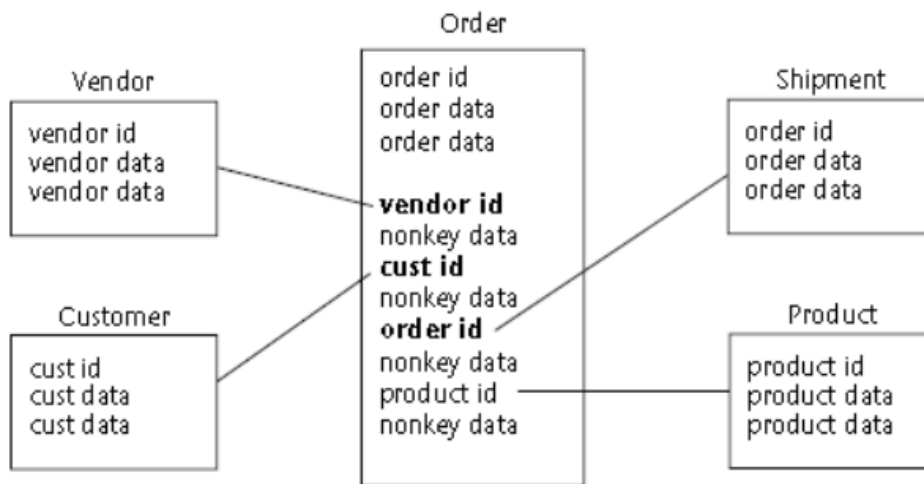
2.9 Pemodelan Multidimensional

Pemodelan Multidimensional adalah sebuah teknik desain logikal yang bertujuan untuk menyajikan *data* dalam bentuk standar, intuitif yang memungkinkan untuk performa akses yang tinggi (Connoly dan Begg, 2005).

2.9.1 *Fact Table* (Tabel Fakta)

Fact table (tabel fakta) adalah pusat dari tabel *star join* dimana *data* mempunyai banyak kejadian yang akan ditempatkan (Inmon : 2005).

UMMN



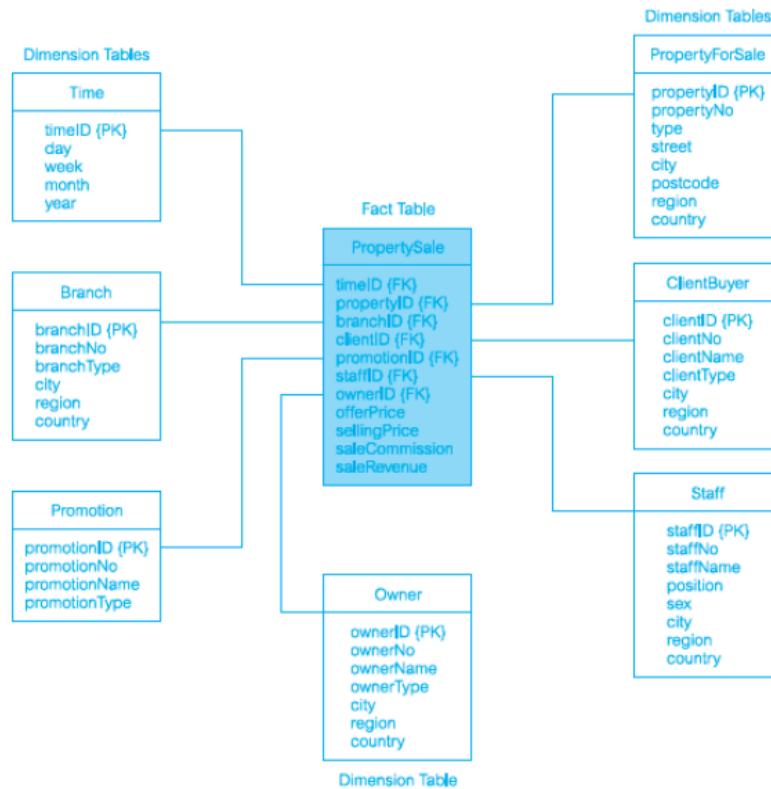
Gambar 2.1 Representasi Entitas dan Relasinya

(Sumber: Connolly dan Begg, 2005)

Fact table merupakan tabel utama yang merupakan inti dari skema. Pada gambar 2.1 pusat dari skema tersebut adalah *order*. *Fact table* mengandung kumpulan *foreign key* dan *primary key* yang ada pada masing-masing tabel dimensi yang berhubungan.

2.9.2 Skema Bintang (*Star schema*)

Skema bintang (*star schema*) adalah struktur logikal yang mempunyai sebuah tabel fakta yang berisi data faktual yang ditempatkan di tengah, dikelilingi oleh tabel dimensi yang berisi data referensi yang dapat dinormalisasi (Connolly dan Begg : 2005)



Gambar 2. 2 Skema Bintang (*Star Schema*)

(Sumber: Connolly dan Begg, 2005)

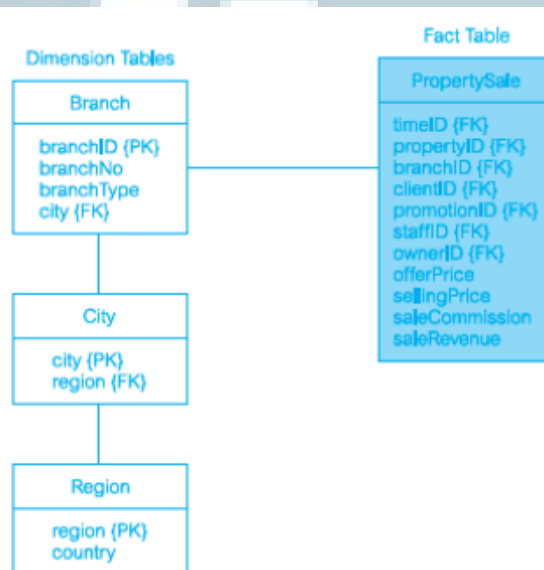
Skema bintang memiliki beberapa keuntungan diantaranya

(Ponniah:2001) :

1. Mudah untuk dimengerti
2. Optimasi navigasi
3. Cocok untuk pemrosesan *Query*
4. Cocok untuk teknik performa yang khusus seperti
5. *STARjoin* dan *STARindex*.

2.9.3 Skema *Snowflake* (*Snowflake schema*)

Skema *snowflake* (*snowflake schema*) adalah variasi lain dari skema bintang yang memungkinkan untuk memiliki banyak dimensi (Connolly dan Begg : 2005). Tabel dimensi dapat mempunyai tabel dimensi lainnya seperti pada gambar 2.3 dibawah ini



Gambar 2. 3 Skema *Snowflake* (*Snowflake Schema*)

(Sumber: Connolly dan Begg, 2005)

Keuntungan dan kerugian dari skema *snowflake* adalah (Ponniah: 2001):

Keuntungan

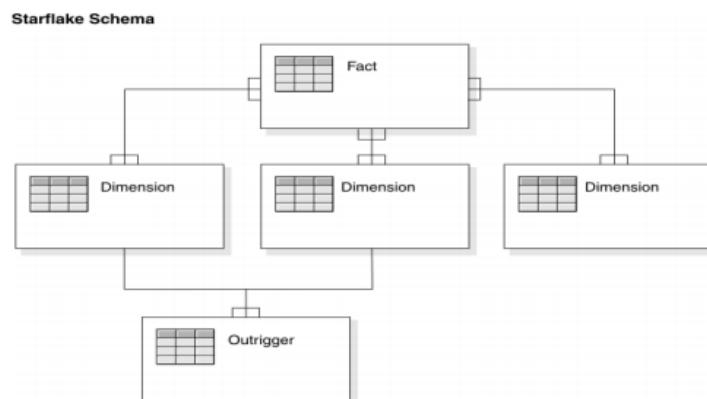
1. Tempat penyimpanan yang kecil
2. Struktur yang telah dinormalisasi akan lebih mudah untuk diupdate dan dikelola

Kerugian

1. Skema kurang intuitif dan *user* ditempatkan pada kompleksitas
2. Sulit untuk menelusuri melalui konten
3. Penurunan performa *query* karena penambahan *join*

2.9.4 Skema *Starflake* (*Starflake Schema*)

Skema *starflake* (*starflake schema*) adalah struktur yang diturunkan dari penggabungan konsep *star schema* dan *snowflake schema*. Beberapa dimensi memiliki kemungkinan dibentuk dengan kedua konsep *Star Schema* dan *Snowflake Schema*, hal ini disesuaikan dengan kebutuhan akan *query* yang dimiliki (Connolly dan Begg : 2005).



Gambar 2. 4 Skema *Starflake* (*Starflake Schema*)

(Sumber: Connolly dan Begg, 2005)

2.10 *Extract Transform Loading*

Extract, Transform, Loading (ETL) adalah serangkaian proses dimana sumber *data* operasional dipersiapkan untuk *data warehouse*. Meliputi ekstraksi *data* operasional dari sumber aplikasi, melakukan transformasi, memuat dan melakukan pengindeksan, kontrol kualitas, dan menerbitkannya (Kimball dan Ross : 2002).

2.10.1 *Extract*

Ekstraksi adalah langkah pertama dalam proses untuk mendapatkan/mengambil *data* untuk dimasukkan ke dalam lingkungan *data warehouse*. Ekstrak berarti membaca dan memahami sumber *data* dan menyalin *data* yang diperlukan untuk *data warehouse* ke dalam *staging area* untuk manipulasi lebih lanjut. *Data* diambil dari satu atau lebih sumber data atau sistem operasional (Kimball dan Ross : 2002).

2.10.2 *Transform*

Transform melakukan proses pembersihan *data* (mengoreksi kesalahan ejaan, mengatasi konflik *domain*, berurusan dengan elemen yang hilang, atau menguraikan ke dalam format standar), menggabungkan *data* dari berbagai sumber, *deduplicating data*, dan menetapkan kunci dari *warehouse* sehingga mendapatkan struktur *data* yang dikehendaki (Kimball dan Ross : 2002).

2.10.3 Loading

Fase *loading* merupakan fase memasukkan *data* ke target akhir, yaitu ke dalam *data warehouse*. Proses ini sangat bervariasi, tergantung pada kebutuhan organisasi Fase *load* berinteraksi dengan *database*, batasan didefinisikan dalam skema *database* sebagai pemicu pada waktu melakukan *load data* (contohnya :*uniqueness, referential, integrity, mandatory fields*), yang juga berkontribusi terhadap kinerja kualitas *data* secara keseluruhan dari proses *ETL* (Kimball dan Ross : 2002).

2.11 MySQL

MySQL adalah perangkat lunak sistem manajemen basis *data SQL* (*database management system*) atau *DBMS* dari sekian banyak *DBMS*, seperti *Oracle, MS SQL, Postagre SQL*, dll (Anhar:2010). *Mysql* adalah *software* atau program *database server* (Bunafit:2013).

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *MySQL* merupakan salah satu perangkat lunak sistem manajemen basis *data SQL* atau *DBMS* yang digunakan pada sisi *server*

2.12 Visualisasi Data

Teknik visualisasi adalah konversi *data* ke dalam format visual atau tabel sehingga karakteristik dari *data* dan relasi antara *item data* atau *attribut* dapat di analisis dan dilaporkan (Asmara:2009). Visualisasi merupakan salah satu cara

untuk merepresentasikan *data* (Hansen dan Johnson: 2005). Bidang visualisasi di fokuskan pada penciptaan gambar yang menyampaikan informasi penting mengenai *data* yang mendasari (Hansen dan Johnson: 2005). Proses pembuatan visualisasi diperlukan sebuah *data*, dimana sumber *data* dapat diperoleh dari hasil simulasi numerik, pengukuran *data* fisik atau *database* (Weiskopf:2006). Teknik visualisasi dapat diterapkan ke dalam sistem informasi atau aplikasi dengan menggunakan koneksi *database* sehingga *data* dapat diproses dan ditampilkan secara dinamis dan *real time* untuk memperluas pemanfaatan *data* (Asmara:2009).

2.13 *Visual Data Mining*

Visual Data Mining (VDM) merupakan metode yang dilakukan untuk melakukan visualisasi *data*. Terdapat 3 fase dan 8 tahap dalam mengerjakan visualisasi *data*, yaitu (Soukup:2002):

A. *Project Planning Phase*

Fase ini merupakan fase perencanaan mengenai kebutuhan *data* yang akan digunakan dan cara pengolahan. Fase ini terdiri atas 2 tahap, yaitu:

UMMN

1. *Justify and Plan the Project*

Langkah ini merupakan penentuan rencana visualisasi yang akan dibuat, cara pembuatan visualisasi dan tujuan dari visualisasi.

2. *Identify the top Business Question*

Langkah ini merupakan proses menentukan pengguna utama dari visualisasi yang akan dibuat.

B. *Data Preparation Phase*

Pada fase ini dilakukan persiapan kebutuhan yang berkaitan dengan kebutuhan yang telah ditentukan pada fase sebelumnya. Fase ini terdiri atas 3 tahap, yaitu:

1. *Choose the Data Set*

Langkah ini merupakan proses menentukan *data* yang akan digunakan dalam melakukan visualisasi *data*

2. *Transform the Data Set*

Langkah ini merupakan proses transformasi dari *data* yang digunakan sesuai dengan kebutuhan proses visualisasi

3. *Verify the Data Set*

Langkah ini merupakan proses pengujian apakah *data* sudah bersih dari bias dan *error*.

C. *Data Analysis Phase*

Pada fase ini dilakukan pemrosesan dan analisa pada *data* yang telah direncanakan dan disiapkan pada fase sebelumnya. Fase ini terdiri atas 3 tahap, yaitu:

1. *Choose the Visualization for Mining Tool*

Langkah ini menentukan *tools* yang akan digunakan dalam proses visualisasi.

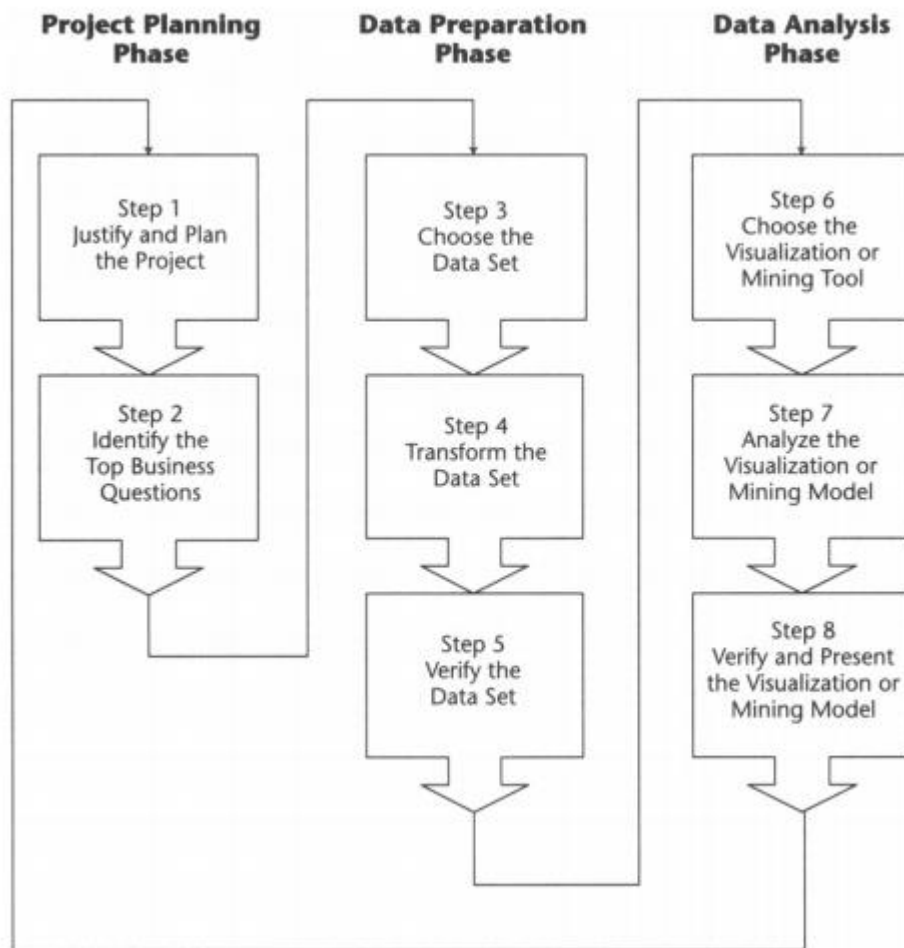
2. *Analyze the Visualization or Mining Model*

Langkah ini melakukan analisa terhadap visualisasi yang telah dibuat.

3. *Verify and Present the Visualization or Mining Model*

Langkah ini melakukan verifikasi terhadap *user* yang akan menggunakan visualisasi untuk memastikan hasil visualisasi sesuai dengan kebutuhan *user*.

U
M
M
N



Gambar 2.5 *Visual Data Mining Phase and Step*

(Sumber: Soukup, 2002)

2.14 *Dashboard*

Dashboard adalah satu kategori dari aplikasi *business intelligence* yang secara *real time* akan memonitoring berbagai informasi yang dibutuhkan oleh suatu organisasi atau perusahaan dengan berbagai macam format seperti *graphical gadgets, typically, gauges, charts, indicators, dan colorcoded maps* yang memungkinkan mereka membuat keputusan pintar secara cepat (Kusnami : 2009).

2.15 Pentaho

Pentaho Kettle menyediakan fasilitas untuk melakukan proses *ETL* (Wibisono: 2012). Terdapat dua elemen utama pada *Pentaho Kettle* yaitu *Transformation* dan *Job*. *Transformation* merupakan sekumpulan instruksi untuk merubah *input* menjadi *output* yang diinginkan, sedangkan *Job* adalah kumpulan instruksi untuk menjalankan proses *transformasi*. Terdapat tiga komponen penyusun aplikasi. Komponen tersebut adalah *Spoon*, *Pan* dan *Kitchen*. *Spoon* merupakan *user interface* untuk membuat *Job* dan *Transformation*, *Pan* digunakan untuk membaca, merubah dan menulis *data*, dan *Kitchen* untuk mengeksekusi *job*.

Pentaho Schema Workbench memberikan grafis antarmuka untuk melakukan perancangan struktur kubus dalam *OLAP* untuk *Pentaho Analysis (Mondrian)*.

Pentaho Community Dashboard Edition (CDE) merupakan *tools* yang terdapat pada *Pentaho BA server* yang digunakan untuk melakukan perancangan *dashboard*. *CDE* mampu digunakan untuk mengembangkan dan menyebarkan *dashboard* dalam *platform pentaho* dengan cepat dan efektif.

2.16 User Acceptance Testing

Tujuan dari pengujian ini adalah mendemonstrasikan bahwa sistem memenuhi persyaratan. Pada pengembangan *software and hardware* komersial, *acceptance test* juga disebut sebagai *alpha* dan *beta test*. *Alpha* dan *beta test* dijalankan untuk mendemonstrasikan kesiapan suatu produk di pasaran (Black : 2009)

2.17 Citra

Citra atau *image* adalah fungsi dua dimensi yang dihasilkan dari penglihatan suatu pemandangan (Schalkoff:1989). Citra adalah kumpulan *pixel-pixel* yang disusun dalam larik dua dimensi, indeks baris dan kolom (x,y) dari sebuah *pixel* dinyatakan dalam bilangan bulat (Ahmad:2005).

2.18 Model Warna Red Green Blue

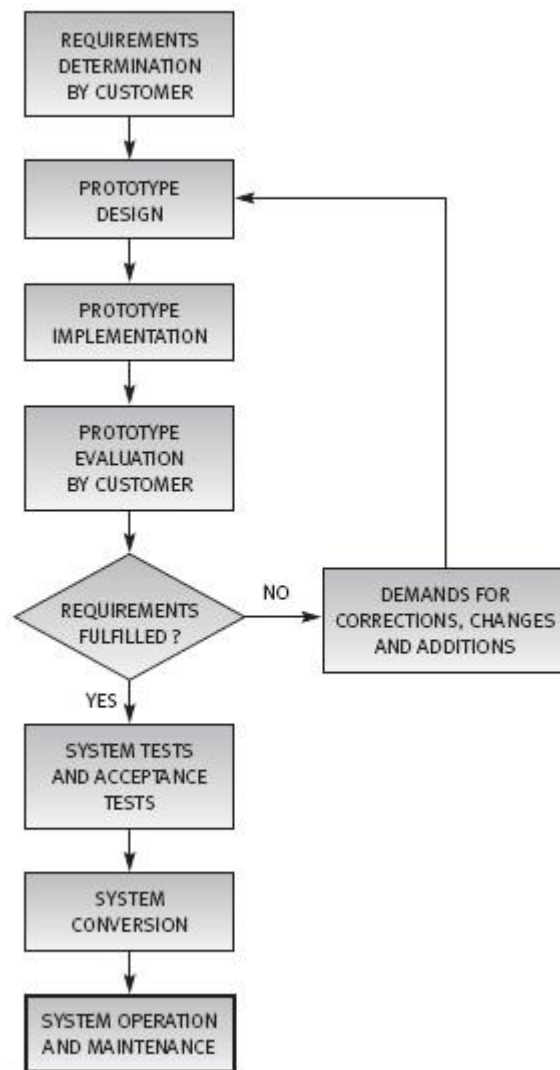
Model warna *Red, Green, Blue (RGB)* adalah model formal untuk menampilkan warna-warna pada monitor komputer, kartu grafis komputer, televisi dan sistem *display* lainnya (Ahmad:2005). Sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*) (Gonzales:1992). Sorotan *electron* menghasilkan sinyal merah, hijau, biru yang dikombinasikan untuk menghasilkan berbagai warna yang dilihat pada layar (Munir:2002).

2.19 Model *Prototype*

Prototipe merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat suatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai. Prototipe membuat proses pengembangan sistem informasi menjadi lebih cepat dan lebih mudah, terutama pada keadaan kebutuhan pemakai sulit untuk diidentifikasi (Kadir:2002).

Tujuan dari penerapan model *prototype* pada pengembangan aplikasi adalah sebagai berikut (Lucas:2000):

1. Mengurangi waktu sebelum pemakai melihat sesuatu yang kongkrit dari usaha pengembangan sistem.
2. Menyediakan umpan balik yang cepat dari pemakai kepada pengembang.
3. Membantu menggambarkan kebutuhan pemakai dengan kesalahan yang lebih sedikit.
4. Meningkatkan pemahaman pengembangan dan pemakai terhadap sasaran yang seharusnya dicapai oleh sistem.
5. Menjadikan keterlibatan pemakai sangat berarti dalam analisis dan desain sistem.



Gambar 2. 6 Prototype Model Diagram

(Galin:2004)

Berikut adalah tahapan dalam metode *prototype* (Werdaya:2012):

1. Komunikasi dan Pengumpulan Data Awal

Melakukan analisis awal terhadap kebutuhan pengguna

2. *Quick Design*

Melakukan pembuatan desain secara umum yang kemudian akan dikembangkan kembali

3. Pembentukan *Prototype*

Proses pembuatan perangkat *prototype* berdasarkan hasil rancangan umum termasuk proses pengujian dan penyempurnaan.

4. Evaluasi terhadap *Prototype*

Melakukan evaluasi terhadap hasil pembuatan *prototype* dan memperhalus analisis terhadap kebutuhan pengguna.

5. Perbaiki *Prototype*

Melakukan perbaikan akhir berdasarkan hasil evaluasi *prototype*

6. Produksi akhir

Melakukan produksi perangkat secara benar sehingga dapat digunakan oleh pengguna

2.20 Model *Rapid Application Development*

Rapid Application Development atau *RAD* adalah sebuah model proses pengembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (Pressman:2002).

Aktivitas-aktivitas pada metode *RAD* adalah sebagai berikut (Pressman, 2005:49):

1. *Communication*

Pada tahap ini digunakan untuk memahami domain masalah dan karakteristik informasi yang harus dihasilkan oleh sistem. Pada tahap ini dilakukan beberapa aktifitas antara lain :

a. *Discover Business Process*

Aktifitas ini bertujuan untuk memahami proses bisnis yang ada

b. *Perform domain analysis*

Dengan menganalisa hasil analisis proses bisnis, kemudian identifikasi entitas-entitas yang memiliki kemungkinan untuk menjadi *class*.

c. *Identify cooperating systems*

Pada tahap awal proses pengembangan sistem dilakukan analisis untuk menentukan hubungan ketergantungan sistem baru dengan sistem yang sudah diterapkan sebelumnya

d. *Discover system requirements*

Pada aktifitas ini tahap *JAD (Joint Application Development)* dilakukan di mana pihak pemegang keputusan memberikan keputusan akhir mengenai kebutuhan sistem yang akan dikembangkan

2. *Planning*

Pada tahap ini ditentukan *domain* masalah dan kebutuhan sistem, batasan sistem dan gambaran fungsi-fungsi yang akan dilakukan sistem.

3. Modelling

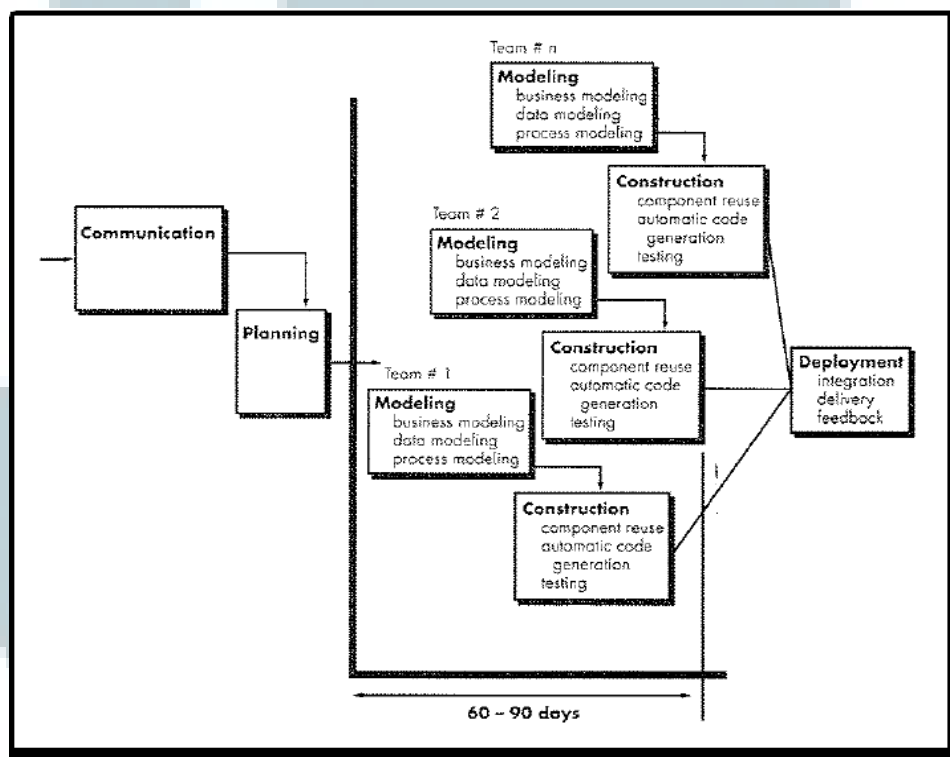
Modeling meliputi 3 (tiga) fase utama pemodelan yaitu bisnis model, *data* model dan proses model. Desain model menggambarkan pemanfaatan dari konstruksi yang akan digunakan.

4. Construction

Aktivitas ini ditekankan pada penggunaan komponen perangkat lunak yang sudah ada dan aplikasi pembuatan kode secara otomatis.

5. Deployment

Pada tahap ini perangkat lunak yang baru dikembangkan akan disempurnakan dan di-*install*. Tahap ini adalah akhir dari proses pengembangan secara *RAD*.



Gambar 2. 7 RAD Model Diagram

(Pressman:2002)