BAB 2
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

CV Mitra Karya Teknik salah satu dari sekian banyak perusahaan yang bergerak di bidang *manufacturing* yang telah berdiri sejak tahun 2001 oleh bapak Ujang Cucu Suryadi. CV Mitra Karya Teknik yang beralamat di Taman Cimekar Kav. 10 no.13 RT 08/RW 11 Ds Cibiru Hilir Kec Cileunyi – Bandung. Dalam menjalankan usahanya CV Mitra Karya Teknik membuat berbagai produk berupa karet, plastic, logam, baklit, nylon, dan polyurethane dll.

2.1.1 Visi Misi

Visi dan misi dari CV Mitra Karya Teknik ini adalah sebagai berikut :

a. Visi

Menjadi perusahaan *manufacturing* (terutama karet) yang memiliki kualitas terbaik Nusantara dan mampu menguasai pasar.

b. Misi

Menjadi perusahaan *manufacture* terbaik Nusantara.

2.1.2 Logo Perusahaan

Logo dari CV Mitra Karya Teknik dapat dilihat pada gambar dibawah ini.

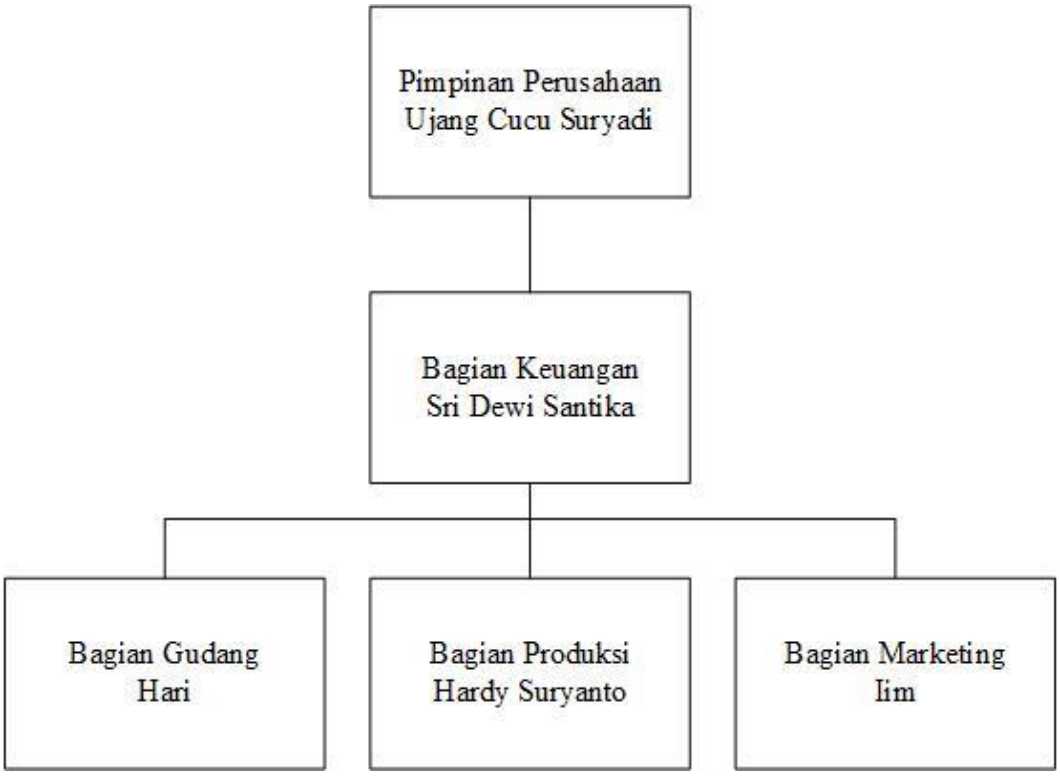


Gambar 2.1 Logo Perusahaan

2.1.3 Struktur Organisasi

Struktur organisasi dalam suatu perusahaan merupakan hal mutlak dan hirarki yang menunjukkan tentang keberadaan jabatan seseorang dalam suatu

perusahaan. Hal ini menyangkut terhadap tanggung jawab mengenai jabatan seseorang dan juga hubungannya terhadap posisi pada jabatan lain **[2]**.



Gambar 2.2 Struktur Organisasi CV Mitra Karya Teknik

Pada struktur organisasi CV Mitra Karya Teknik memiliki masing – masing tugas adalah :

1. Tugas pimpinan perusahaan adalah memberikan kerangka pokok yang jelas yang dapat dijadikan pegangan oleh anggotanya, mengawasi, merencanakan, mengorganisasikan, menggerakan anggotanya, dan bertanggung jawab untuk pengambilan keputusan bagi perusahaan.
2. Tugas bagian keuangan adalah melaksanakan kegiatan administrasi keuangan perusahaan yang bertujuan untuk menjamin kelancaran, keakuratan, dan ketertiban keuangan perusahaan itu sendiri.
3. Tugas bagian gudang adalah sebagai penanggung jawab dari semua transaksi yang terjadi di bagian gudang perusahaan tersebut.
4. Tugas bagian produksi adalah membuat produk dari mulai mengolah bahan mentah menjadi barang jadi.

5. Tugas bagian marketing dan penjualan adalah mengatur jadwal penjualan dan melakukan penjualan untuk memenuhi permintaan pelanggan serta mencatat semua transaksi penjualan.

1.2 Landasan Teori

Landasan teori merupakan teori yang relevan yang digunakan untuk menjelaskan tentang *variabel* yang akan diteliti dan sebagai dasar untuk memberi jawaban sementara terhadap rumusan masalah yang diajukan (*hipotesis*), dan penyusunan instrument penelitian. Landasan teori yang diuraikan merupakan hasil dari studi literatur yaitu pengumpulan data dengan cara mempelajari jurnal, browing internet, referensi buku, dan bacaan – bacaan lainnya yang berhubungan dengan penelitian yang dilakukan.

2.2.1 Data

Data adalah kumpulan kejadian yang diangkat dari suatu kenyataan (fakta), dapat berupa angka – angka, huruf, simbol – simbol khusus, atau gabungan dari ketiganya.

Data juga bisa berarti kumpulan file atau informasi dengan tipe tertentu, baik suara, gambar atau yang lainnya. Data merupakan keterangan yang benar dan nyata, atau keterangan nyata yang dapat dijadikan bahan kajian (analisis atau kesimpulan) **[3]**.

2.2.2 Database

Database adalah himpunan kelompok data (arsip) berupa file, table, dan arsip yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah dan saling berhubungan yang disimpan secara bersamaan sedemikian rupa tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan yang disimpan dalam media penyimpanan elektronis **[3]**.

2.2.3 Database Management System (DBMS)

Database management sistem (DBMS) adalah sistem pengorganisasian dan sistem pengolahan database pada komputer. DBMS ini merupakan perangkat lunak (*software*) yang dipakai untuk membangun basis data yang berbasis komputerisasi. DBMS ini juga dapat membantu dalam memelihara serta pengolahan data dalam jumlah yang besar, dengan menggunakan DBMS bertujuan agar tidak dapat menimbulkan kekacauan dan dapat dipakai oleh user sesuai dengan kebutuhan **[3]**.

2.2.4 Data Warehouse

Pengertian *data warehouse* dapat bermacam – macam namun mempunyai inti yang sama, seperti pendapat beberapa ahli berikut ini **[4]**:

1. *Data warehouse* adalah kumpulan data yang memiliki subjek berorientasi, terpadu, waktu – variant, dan tetap pada pengumpulan data untuk mendukung proses pengambilan keputusan management (*W.H. Inmon dan Richard DH*).
2. *Data warehouse* adalah database yang *read – only* analisis dan digunakan sebagai dasar sistem pendukung (*Vidette Poe*).
3. *Data warehouse* merupakan database relasional yang dirancang lebih untuk *query* dan analisis proses transaksi, biasanya mengandung sejarah transaksi dan mungkin juga data dari sumber lain. *Data warehouse* memisahkan beban kerja *analisis* dari beban kerja transaksi dan memungkinkan organisasi untuk menggabungkan konsolidasi data dari berbagai sumber (*Paul Lane*).
4. *Data warehouse* merupakan metode dalam perancangan database yang menunjang DSS (*Executive Information System*) dan EIS (*Executive Information System*). Secara fisik *Data Warehouse* adalah database, akan tetapi perancangan *data warehouse* dan database sangat berbeda. Dalam desain database tradisional menggunakan normalisasi, sedangkan *data warehouse* normalisasi bukanlah cara terbaik.

Dari definisi yang sudah diuraikan diuraikan di atas, dapat disimpulkan *data warehouse* adalah *database* yang saling berinteraksi yang dapat digunakan untuk *query* dan *analisis*, bersifat *orientasi subjek*, *terintegrasi*, *time – variant*, dan *multidimensi* tidak berubah yang digunakan untuk pimpinan perusahaan untuk pengambilan keputusan.

2.2.4.1 Kegunaan Data Warehouse

Dari pengertian yang sudah diuraikan tentang *data warehouse, data warehouse* diperlukan bagi para pimpinan perusahaan untuk pengambilan keputusan management dari suatu organisasi/perusahaan. Dengan adanya *data warehouse* akan mempermudah pembuatas DSS (*Dicision Support System*) dan EIS (*Executive Information System*) karena kegunaan dari *data warehouse* adalah khusus untuk membuat suatu *database* yang digunakan untuk mendukung proses analisa (OLAP), pengambilan keputusan, pembuatan laporan, penggalian informasi baru (*Data Mining*) dari banyak data dan proses *executive information* Beberapa fungsi dari *warehouse* diantaranya **[4]** :

1. Pembuatan Laporan

   Pembuatan laporan merupakan salah satu kegunaan *data warehouse* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan perhari, perbulan, bahkan sampai pertahun atau jangka waktu kapanpun yang diinginkan.

2. OLAP (*Online Analytical Processing*)

   Dengan adanya data warehouse, semua informasi baik detail maupun summary yang dibutuhkan dalam proses analisa mudah didapat. OLAP berfungsi untuk mengakses yang besar, OLAP berguna untuk menganalisis hubungan antar data guna menemukan ada tidak pola tertentu dalam data yang berguna untuk pengambilan keputusan di masa yang akan datang. Fasilitas lain yang ada pada sofware OLAP adalah fasilitas rool-up dan drill-down. Drill-down adalah kemampuan untuk melihat detail dari suatu informasi dan roll-up adalah kebalikannya.

3. Data Mining

   Data mining merupakan proses untuk menggali (mining) pengetauan pengetahuan dan informasi baru dari data yang berjumlah banyak pada data warehouse, dengan menggunakan kecerdasan buatan (Artificial Intelegence), statistik dan matematika. Data mining merupakan teknologi yang diharapkan dapat menjembatani komunikasi data dan pemakainya.

   Adapun beberapa solusi yang diberikan oleh *data mining* antara lain :

   1. Menebak Target Pasar

      Data mining dapat mengelompokkan (*clustering*) model-model pembeli dan melakukan klasifikasi terhadap setiap pembeli dan melakukan klasifikasi terhadap setiap pembeli sesuai dengan karakteristik yang diinginkan.

   2. Melihat Pola Beli dari Waktu ke Waktu

      Data mining dapat digunakan untuk melihat pola beli dengan rentang waktu ke waktu.

   3. *Cross-market Analysis*

      Data mining dapat dimanfaatkan untuk melihat hubungan antara satu produk dengan produk lainnya.

   4. Profil Pelanggan

      Data mining bias membantu pengguna untuk melihat profil pembeli sehingga dapat diketahui kelompok pembeli tertentu hanya cendrung pada produk saja.

   5. Informasi *Summary*

      Data mining dapat membuat laporan summary yang bersifat *multidimensi* dan dilengkapi dengan *statistic*.

   6. Proses Informasi Executive

      Data warehouse dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data.

2.2.4.2 Konsep Dasar Data Warehouse

Data warehouse adalah kumpulan macam – macam data yang *subject oriented, integrated, time variant,* dan *multidimensi* dalam mendukung proses keputusan yang dibuat oleh management perusahaan.

Data warehouse sering sering *diintegrasikan* dengan berbagai sistem aplikasi untuk mendukung proses laporan dan analisis data dengan menyediakan *data histori*, yang menyediakan infrastruktur bagi EIS dan DSS **[4]**.

1. *Integrated*

   Sebuah data warehouse memiliki data yang diambil dari berbagai sistem operasional dan data eksternal.

2. *Subject Oriented*

   Data warehouse data diperoleh berdasarkan enterprise subject seperti pelanggan, sales, dan profit.

3. *Time Variant*

   Seluruh data pada data warehouse dapat dikatakan akurat atau valid pada rentang waktu tertentu. Untuk melihat interval waktu yang digunakan dalam mengukur keakuratan suatu data warehouse.

4. *Non Volatile*

   Setiap kali proses perubahan, data akan di tampung dalam tiap-tiap waktu, jadi tidak di perbaharui terus menerus. Data warehouse tidak memerlukan pemrosesan transaksi dan recovery. Hanya ada dua operasi initial loading of data dan *access of data*.

2.2.4.3 Proses ETL (*Extraction, Transformation, Loading*)

Extraction, transformation, dan *loading* (ETL) adalah fungsi utama yang perlu dilakukan untuk membuat data. Ketiga fungsi ini terdapat pada *staging area*. Pada *data staging* ini, disediakan tempat dan area dengan beberapa fungsi seperti *data cleansing, charge, convert,* dan menyiapkan data untuk disimpan serta digunakan dalam *data warehouse* **[4]**.

1. *Extraction*

   Data *extraction* adalah proses pengambilan data yang diperlukan dari sumber data warehouse dan selanjutnya dimasukkan pada staging area untuk diproses pada tahap berikutnya. Pada fungsi ini banyak berhubungan dengan berbagai tipe sumber data seperti format data, mesin yang berbeda, software dan arsitektur yang tidak sama. Sehingga sebelum proses ini dilakukan, sebaiknya perlu didefinisikan requirement terhadap sumber data yang yang akan digunakan untuk proses berikutnya. Adapun fungsi ekstrasi diantaranya, yaitu :

   a. *Ekstraksi* data secara otomatis dari aplikasi sumber.
   b. Penyaringan atau seleksi data hasil *ekstraksi*.
   c. Pengiriman data dari berbagai *platform* aplikasi ke sumber data.
   d. Perubahan format *layout* data dari format aslinya.
   e. Penyimpanan dalam *file* sementara untuk penggabungan dengan hasil *ekstraksi* dari sumber lain.

2. *Tranformation*

   Pada kenyataannya, pada proses *transaksional* data disimpan dalam berbagai format sehingga jarang kita temui data yang konsisten antara aplikasi-aplikasi yang ada. *Transformasi* data ditujukan untuk mengatasi masalah ini. Dengan proses *transformasi* data ini, kita melakukan standarisasi terhadap data pada satu format yang konsisten. Beberapa contoh ketidak konsistenan data tersebut dapat diakibatkan oleh tipe data yang berbeda, data length dan lain sebagainya.. Langkah-langkah dalam *transformasi* data adalah sebagai berikut :

   a. Memetakan data *input* dari skema data aslinya ke skema *data warehouse*.
   b. Melakukan *konversi* tipe data atau format data.
   c. Pembersihan serta pembuangan duplikasi dan kesalahan data.

d. Penghitungan nilai-nilai *derivat* atau mula-mula.
e. Penghitungan nilai-nilai *agregat* atau rangkuman.
f. Pemerikasaan *integritas* referensi data.
g. Pengisian nilai-nilai kosong dengan nilai *default*.
h. Penggabungan data.

3. *Loading*

   Data *load* adalah memindahkan data ke *data warehouse*. Ada dua *loading* data yang dilakukan pada *data warehouse*. Pertama adalah *inisial load*, proses ini dilakukan pada saat telah selesai mendisain dan membangun *data warehouse*. Data yang dimasukkan akan sangat besar dan memakan waktu yang relatif lebih lama. Kedua *Incremental load*, dilakukan ketika *data warehouse* telah dioperasikan. *Incremental load* ini dapat dilakukan sesuai dengan sistem yang dibangun.

2.2.4.4 OLTP (On-line Transaction Processing)

On-line Transaction Processing atau yang sering dikenal dengan OLTP adalah segala penanganan dalam proses transaksi sehari – hari. Ciri – ciri dari OLTP adalah sebagai berikut [13] :

1. Mendukung jumlah pengguna dalam jumlah yang banyak dan sering untuk menambah dan mengubah data transaksi.
2. Mengandung data dalam jumlah besar, termasuk di dalamnya validasi data transaksi.
3. Memiliki struktur yang kompleks dan rumit.
4. Diarahkan secara maksimal untuk melayani aktivitas transaksi harian.
5. Menyediakan teknologi infrastruktur yang mendukung operasional transaksi data dalam perusahaan

2.2.4.5 OLAP (*Online Analitical Proses*)

OLAP merupakan salah satu katagori teknologi perangkat lunak yang memungkinkan analisis, manager, dan eksekutif untuk menggali suatu data secara

cepat, konsisten, dan memiliki akses yang interaktif dalam luasnya penggalian suatu informasi yang di *tranformasikan* dari suatu *raw* data menjadi dimensi fakta yang dapat dimengerti oleh pengguna **[13]**.

1. Slicing dan Dicing

   Slicing dan dicing adalah operasi untuk melihat data sebagai visualisasi dari kubus. Dengan slicing dan dicing pengguna dapat melihat data dari beberapa perspektif. Pengguna dapat mengekstrak bagian dari data agregrated dan dapat memeriksa dengan detail berdasarkan dimensi-dimensi yang diinginkan. Data *Agregrated* merupakan data praperhitungan (*precalculated*) dalam bentuk rangkuman data (data summarized) sehingga query pada kubus (*cube*) lebih cepat. *Slicing* memotong kubus sehingga dapat memfokuskan pada perspektif yang spesifik. Sedangkan dicing memberikan kemampuan untuk melihat pemilihan data pada dua dimensi atau lebih. Yaitu dengan merotasi cube pada perspektif yang lain sehingga pengguna dapat melihat lebih spesifik terhadap data yang dianalisa.

2. *Roll-up* dan *Drill down*

   Roll-up dan d*rill down* adalah operasi untuk melihat data global atau detail disepanjang level hiraraki dimensi. *Roll up* untuk melihat data secara global atau rangkuman (summary). *Drill down* memandu pengguna untuk memperoleh data yang lebih detail. *Drill down* ini biasa digunakan untuk menjawab pertanyaan atas suatu kasus tertentu. Misalnya untuk menjawab pertanyaan ketika sebuah summary number (rata-rata atau jumlah) di bawah atau di atas harapan.

2.2.4.6 Normalisasi Data

Normalisasi merupakan teknik analisis data yang mengorganisasikan atribut-atribut data dengan cara mengelompokkan sehingga terbentuk entitas yang non-redundant, stabil, dan fleksible. Normalisasi dilakukan sebagai uji coba pada suatu relasi secara berkelanjutan untuk menentukan apakah relasi itu sudah baik,

yaitu dapat dilakukan proses insert, update, delete, dan modifikasi pada satu atau beberapa atribut tanpa mempengaruhi integritas data dalam relasi tersebut.

Pada proses normalisasi terhadap tabel pada database dapat dilakukan dengan tiga tahap normalisasi antara lain [13]:

1. Bentuk Tidak Normal (*Unnormalized Form*)

   Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikukti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan saat menginput.

   Untuk mentransformasikan tabel yang belum ternomalisasi di atas menjadi tabel yang memenuhi kriteria 1NF adalah kita harus merubah seluruh atribut yang multivalue menjadi atribut single value, dengan cara menghilangkan reporting group.

2. Bentuk Normal Ke Satu (*First* Normal form/1NF)

   Pada tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi di antara setiap baris pada suatu tabel, dan setiap atribut harus mempunyai nilai data yang atomic (bersifat atomic value). Atom adalah zat terkecil yang masih memiliki sifat induknya, bila terpecah lagi maka ia tidak memiliki sifat induknya. Syarat normal ke satu (1-NF) antara lain :

   a. setiap data dibentuk dalam flat file, data dibentuk dalam satu record demi satu record nilai dari field berupa “atomic value”.
   b. tidak ada set atribute yang berulang atau bernilai ganda.
   c. telah ditentukannya primary key untuk tabel / relasi tersebut.
   d. tiapatribut hanya memiliki satu pengertian.

3. Bentuk Normal Ke Dua (*Second* Normal Form/2NF)

   Bentuk normal kedua didasari atas konsep full functional dependency (ketergantungan fungsional sepenuhnya) yang dapat didefinisikan sebagai berikut. Jika A adalah atribut-atribut dari suatu relasi, B dikatakan full functional dependency (memiliki ketergantungan fungsional terhadap A, tetapi tidak secara tepat memiliki ketergantungan fungsional dari subset (himpunan bagian) dari A. Syarat normal kedua (2/NF) sebagai berikut :

a. Bentuk data telah memenuhi kriteria bentuk normal kesatu.
b. Atribute bukan kunci (non-key) haruslah memiliki ketergantungan fungsional sepenuhnya (fully functional dependency) pada kunci utama / primary key.

4. Bentuk Normal Ke Tiga (*Third* Normal Form/3NF)

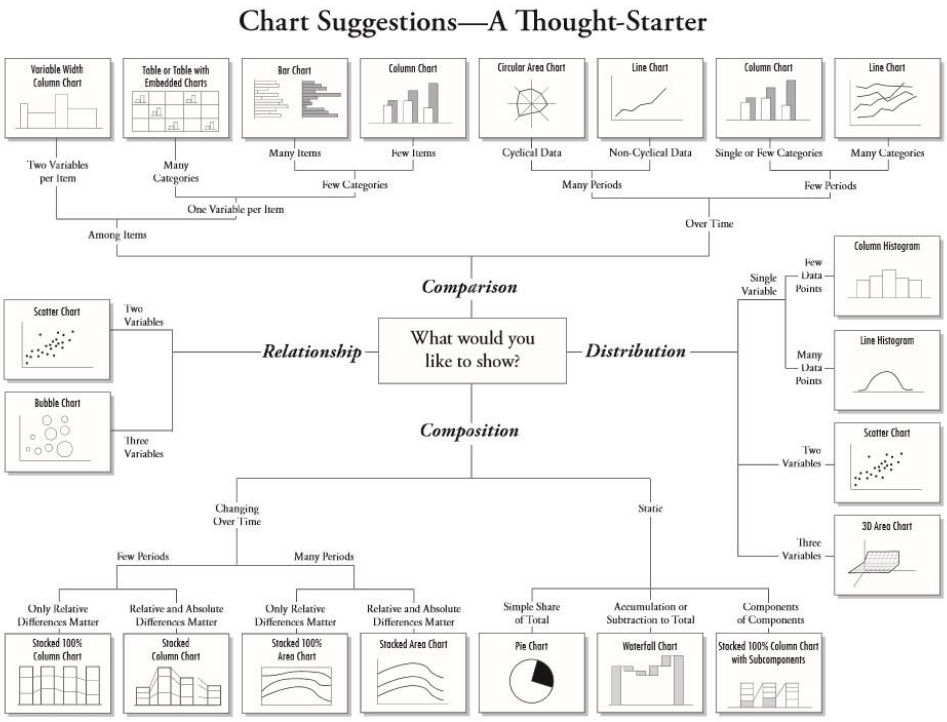
   Walaupun relasi 2-NF memiliki redudansi yang lebih sedikit dari pada relasi 1-NF, namun relasi tersebut masih mungkin mengalami kendala bila terjadi anomaly peremajaan (update) terhadap relasi tersebut. Misalkan kita akan melakukan update terhadap nama dari seorang Pemilik (pemilik), seperti Durki (No_Pemilik: CO93), kita harus melakukan update terhadap dua baris dalam relasi Property_Pemilik (lihat Tabel 9.5, (c) relasi Property_Pemilik). Jika kita hanya mengupdate satu baris saja, sementara baris yang lainnya tidak, maka data didalam database tersebut akan inkonsisten / tidak teratur. Anomaly update ini disebabkan oleh suatu ketergantungan transitif (transitive dependency). Kita harus menghilangkan ketergantungan tersebut dengan melakukan normalisasi ketiga (3-NF). Syarat normal ketiga (Third Normal Form / 3 NF) sebagai berikut :

   a. Bentuk data telah memenuhi kriteria bentuk normal kedua.
   b. Atribute bukan kunci (non-key) harus tidak memiliki ketergantungan transitif, dengan kata lain suatu atribut bukan kunci (non_key) tidak boleh memiliki ketergantungan fungsional (functional dependency) terhadap atribut bukan kunci lainnya, seluruh atribut bukan kunci pada suatu relasi hanya memiliki ketergantungan fungsional terhadap priamry key di relasi.

2.2.4.7 Visualisasi Data

Pengertian Visualisasi adalah rekayasa dalam pembuatan gambar, diagram, atau animasi untuk menampilkan sebuah informasi. Secara umum, visualisasi digambarkan dalam bentuk gambar yang bersifat abstrak maupun

nyata. Contoh dari hal ini meliputi lukisan di dinding – dinding gua dari manusia purba, bentuk huruf hiroglip Mesir, sistem geometri Yunani, dan lainnya**[12]** :



Gambar 2.3 *Chart Suggestion*

Pada saat ini visualisasi telah berkembang dan banyak dipakai untuk keperluan ilmu pengetahuan, rekayasa, visualisasi desain produk, pendidikan, multimedia interaktif dll. Visualisasi data memiliki tujuan yaitu :

1. Mengeksplor
2. Menghitung
3. Menyampaikan

Data numerik, skema, gambar umum, tabel, dan lainnya dapat divisualisasikan dalam bentuk media dua dimensi non projeksi. Yang biasa digunakan antara lain :

1. Grafik

   Grafik adalah visualisasi data yang menggambarkan hubungan numerik antara dua variabel. Macam – macam grafik antara lain grafif garis (*line graph*), grafik batang (*bar graph*), grafik lingkaran

(*circle*), grafik luasan (*area graph*), grafik solid (*solid graph*), dan grafik piktorial (*pictorial graph*).

2. Diagram

   Diagram adalah berkas garis dan simbol yang dirancang untuk menunjukan hubungan, gambaran umum, atau ringkasan suatu proses.

3. Peta (*Chart*)

   Peta (*chart*) yang biasa juga diistilahkan karta, merupakan kombinasi dari piktorial, grafik, numerik, atau material verbal. Macam – macam peta (*chart*) peta pohon (*free chart*), peta arus (*flow chart*), peta garis besar (*outine chart*), dan peta tabulasi (*tabular chart*).

4. Kurtun

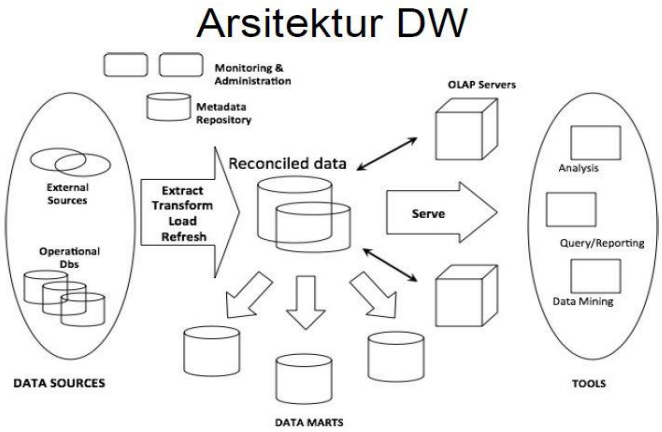
   Kurtur adalah gambaran piktorial karikatur, simbolise, dan humor.

2.2.4.8 Arsitektur Data Warehouse

Arsitektur *data warehouse* adalah sekumpulan aturan atau struktur yang menyediakan sebuah *framework* untuk keseluruhan desain dari sistem atau produk. Di dalam arsitektur terdapat arsitektur jaringan, arsitektur *client-server*, arsitektur untuk produk spesifik, dll. Karakteristik arsitektur *data warehouse* adalah sebagai berikut **[5]** :

a. Data diekstrak dari *source system*, *database,* dan file.
b. Data dari *source system* diintegrasikan dan ditranformasikan sebelum *diload* kedalam *data warehouse*.
c. *Data warehouse* adalah *database read-only* yang terpisah yang dibuat khusus untuk DSS (*decision support system*).
d. *User* dapat mengakses *data warehouse* melalui *tool front end* dan aplikasi.

Arsitektur *data warehouse* mencakup proses ETL (*Extraction, Transformation, Loading*) untuk memindahkan data dari operational *data source* dan sumber data *eksternal* lainnya ke dalam *data warehouse.*



Gambar 2.4 Data Flow Architecture

Berikut adalah penjelasan dari tiap proses :

1. Ekstraksi Data (*Extract*)

   Ekstraksi data adalah proses dimana data diambil atau diekstrak dari berbagai sistem operasional, baik menggunakan *query*, atau aplikasi ETL. Terdapat beberapa fungsi ekstraksi data, yaitu :

   a. Ekstraksi data secara otomatis dari aplikasi sumber.
   b. Penyaringan atau seleksi data hasil ekstraksi.
   c. Penyaringan atau seleksi data hasil ekstraksi.
   d. Perubahan format layout data dari format aslinya.
   e. Penyimpanan dalam file sementara untuk penggabungan dengan hasil ekstraksi dari sumber lain.

2. Transformasi Data (*Transformation*)

   Transformasi adalah proses dimana data mentah (*row data*) hasil ekstrasi disaring dan diubah sesuai dengan kaidah bisnis yang berlaku. Langkah – langkah dalam transformasi data adalah sebagai berikut :

   a. Memetakan data input dari skema data aslinya ke skema *data warehouse.*
   b. Melakukan konversi tipe data atau format data.
   c. Pembersihan serta pembuangan duplikasi dan kesalahan data.
   d. Penghitungan nilai-nilai *derivat* atau mula-mula.

e. Penghitungan nilai-nilai agregat atau rangkuman.

f. Pemerikasaan integritas referensi data.

g. Pengisian nilai-nilai kosong dengan nilai *default*.

h. Penggabungan data.

3. Proses Data (*Loading*)

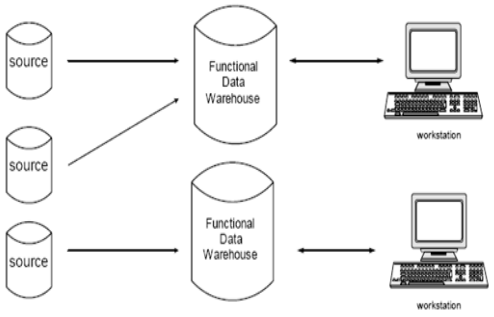
   Proses terakhir yang perlu dilakukan adalah proses pemuatan laporan data yang didapatkan dari hasil transformasi ke dalam *data warehouse*. Cara untuk memuat data adalah dengan menjalankan *SQLscript* secara periodic.

2.2.4.9 Jenis Dasar Data Warehouse

Jenis *data warehouse* yang digunakan oleh sebuah perusahaan akan berbeda-beda, *data warehouse* yang dibangun harus disesuaikan dengan sumber data dan kebutuhan perusahaan tersebut. Berikut ini adalah *jenis data warehouse* :

1. *Functional Data Warehouse* (Data Warehouse Fungsional)

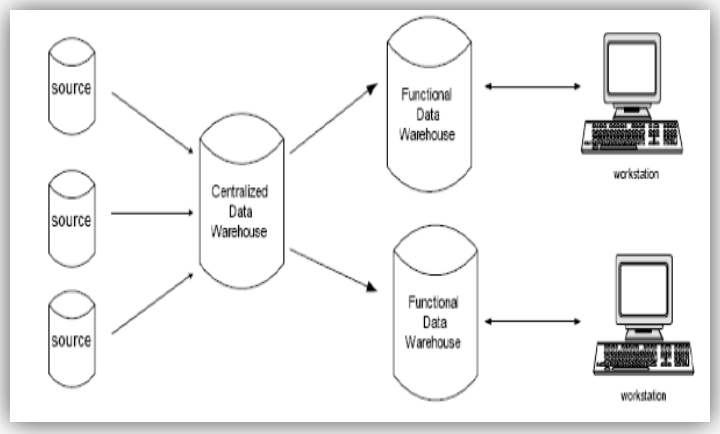
   Data base yang digunakan jenis *data warehouse* ini adalah *database* yang diperoleh dari kegiatan sehari-hari. *Data warehouse* dibuat lebih dari satu dan dikelompokkan berdasar fungsi - fungsi yang ada di dalam perusahaan seperti fungsi keuangan (*financial*), marketing, personalia dan lain-lain. Keuntungannya dari *warehouse* seperti ini adalah sistem mudah dibangun dengan biaya relatif murah sedangkan kerugiannya adalah resiko kehilangan konsistensi data dan terbatasnya kemampuan dalam pengumpulan data bagi penggunanya.



Gambar 2.5 *Data Warehouse Fungsional*

2. *Centralized Data Warehouse* (Data Warehouse Terpusat)

   Bentuk ini terlihat seperti bentuk *data warehouse fungsional*, namun terlebih dahulu sumber data dikumpulkan dalam satu tempat terpusat, kemudian data disebar ke dalam fungsinya masing-masing, sesuai kebutuhan perusahaan. *Data warehouse* terpusat ini, biasa digunakan oleh perusahaan yang belum memiliki jaringan *eksternal*. Keuntungan dari bentuk ini adalah data benar-benar terpadu karena konsistensinya yang tinggi, sedangkan kerugiannya adalah biaya yang mahal serta memerlukan waktu yang cukup lama untuk membangunnya.

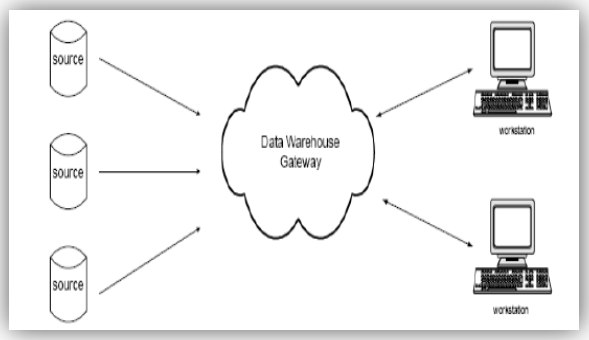


Gambar 2.6 *Data Warehouse* Terpusat

3. *Distributed Data Warehouse* (Data Warehouse terdistribusi)

   Pada *data warehouse terdistribusi* ini, digunakan *gateway* yang berfungsi sebagai jembatan penghubung antara *data warehouse* dengan *workstation* yang menggunakan sistem beraneka ragam. Dengan sistem terdistribusi seperti ini memungkinkan perusahaan dapat mengakses sumber data yang berada di luar lokasi perusahaan (*eksternal*). Keuntungannya adalah data tetap bersifat konsisten karena sebelum data digunakan, data terlebih dahulu disesuaikan atau mengalami proses sinkronisasi. Sedangkan kerugiannya adalah lebih kompleks untuk diterapkan karena sistem operasi dikelola

secara terpisah, juga biayanya yang paling mahal dibandingkan dengan dua bentuk *data warehouse* lainnya.



Gambar 2.7 *Data Warehouse* Terdistribusi

2.2.4.10 Metodologi Pembangunan Data Warehouse

Sembilan tahap metodologi dalam perancangan database untuk *data warehouse*, yaitu **[10]**:

1. Langkah 1 : Pemilihan Proses Bisnis

   Tahapan ini berarti memahami dan memilih proses bisnis mana saja yang akan digunakan dalam model data multidimensi yang akan dibangun.

2. Langkah 2 : Pemilihan Sumber

   Dalam tahap ini, perancangan data warehouse harus sesuai atau berdasarkan dari tabel fakta dan dimensi, untuk itu ketika pemilihan sumber harus mengacu pada kemampuan unutk merepresentasikan fakta-fakta yang di ambil dari berbagai sumber dimensi pada OLTP.

3. Langkah 3 : Mengidentifikasi Dimensi dan Fakta

   Dimensi digunakan untuk menjelaskan entitas bisnis yang digunakan untuk mendukung suatu fakta sehingga dapat dilakukan analisis secara multidimensional.

4. Langkah 4 : Data Staging

   Pada lapisan ini, data ekternal yang sudah diimport kedalam database akan diekstrak, *ditransform* dan kemudian *diload* ke dalam *data warehouse*. Proses ini lebih dikenal dengan proses ETL.

5. Langkah 5 : Analisis Dimensi

   Dimensi digunakan untuk menjelaskan entitas bisnis yang digunakan untuk mendukung suatu fakta sehingga dapat dilakukan analisis secara multidimensional.

6. Langkah 6 : Analisis Fakta

   Fakta menjelaskan tentang nilai yang berkaitan langsung dengan proses bisnis yang bersifat multidimensional dengan maksud agar pengguna dapat lebih mudah untuk melakukan analisis.

7. Langkah 7 : Data Warehouse Layer

   Pada lapisan ini, data yang sudah melalui proses ETL akan disimpan pada sebuah penyimpanan logic yang tersentralisasi yaitu *data warehouse.*

8. Langkah 8 : OLAP

   Setelah *data warehouse* terbentuk, langkah terakhir adalah melakukan pengambilan data dari *data warehouse*. Dalam pembangunan *data warehouse* kali ini, hasil *output* dari *data warehouse* akan berupa laporan dan juga digunakan untuk analisis data dengan OLAP.

9. Langkah 9 : Visualisasi Data

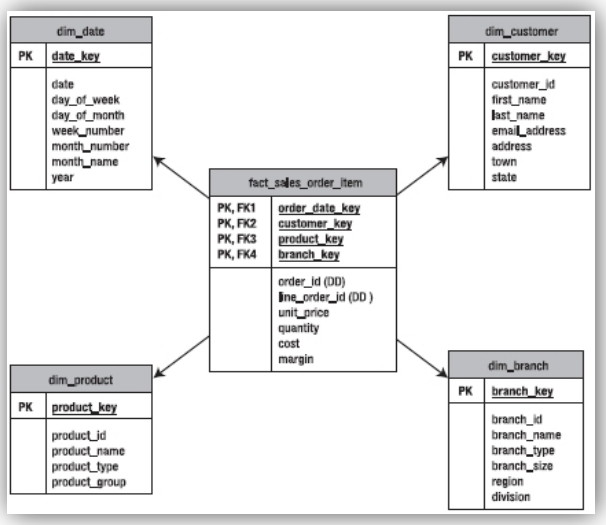
   Dalam tahapan ini kebutuhan informasi strategis yang ada disajikan dalam bentuk halaman antarmuka berupa visualisasi data ke dalam bentuk *bar chart* dan *circle chart* yang digunakan untuk menggambarkan peforma bisnis yang digunakan untuk dapat mempermudah management perusahaan dalam membaca atau mendapatkan informasi secara cepat.

2.2.4.11 Skema Data Warehouse

Schema adalah kumpulan objek-objek database, mencakup tabel, view, index, dan sinonim ada beberapa cara dalam menyusun objek-objek schema dalam model skema yang dirancang untuk *data warehouse*, skema yang sering digunakan pada *data warehouse* adalah *star schema*, *snowflake*, kedua skema ini sangat mudah dimengerti dan sesuai dengan kebutuhan bisnis, mendukung *query* sederhana dan menyediakan performa *query* yang superior dengan meminimalisasi table-tabel join **[5]**.

1. *Star Schema*

   Disebut *star schema* karena *entity-relationship diagram* atau ERD yang menyerupai konstelasi bintang, beberapa bintang besar (*fact table*) dikelilingi bintang - bintang yang lebih kecil (*dimension tables*). Berikut adalah contoh gambar star schema :

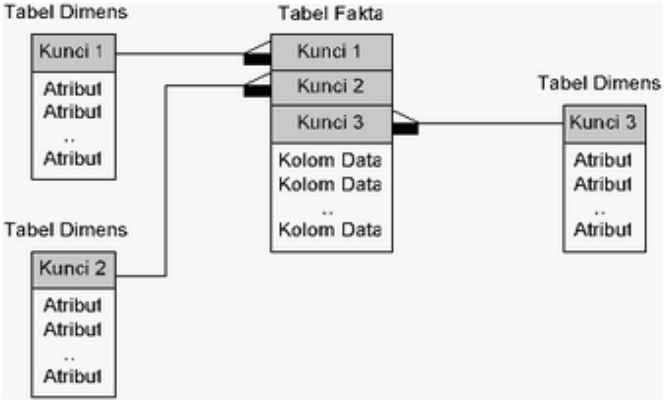


Gambar 2.8 *Star Schema*

Jenis – jenis *star schema* antara lain :

a. Skema bintang sederhana

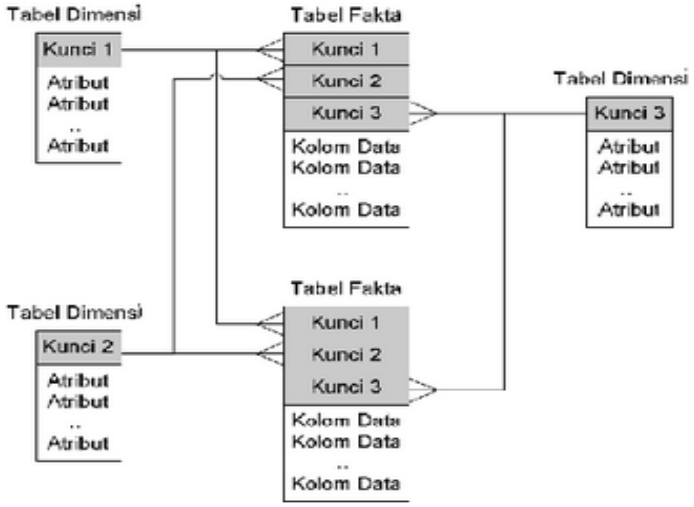
   Dalam skema ini, setiap table harus memiliki *primary key* yang terdiri dari satu kolom atau lebih. *Primary key* dari table fakta terdiri dari satu atau lebih *foreign key*. *Foreign key* merupakan *primary key* pada table lain.



Gambar 2.9 Skema Bintang Sederhana

b. Skema Bintang dengan Banyak Tabel Fakta

Dalam sebuah skema bintang, dapat juga memiliki lebih dari satu table fakta, karena adanya fakta yang saling berhubungan, misalnya disamping penjualan terdapat table fakta *forecasting* dan table fakta *result*. Tetapi walaupun terdapat banyak table fakta, table dimensinya tetap digunakan secara bersama – sama. Gambar di bawah ini menunjukkan adanya dua table fakta dan tiga table dimensi yang memperlihatkan hubungan *many to one* antara *foreign key* pada kedua table fakta tersebut dengan *primary key* pada masing – masing table dimensi. Berikut gambar skema bintang dengan banyak table fakta :



Gambar 2.10 Skema Dengan Banyak Tabel Fakta

2. *Snowflake Schema*

Sedangkan *snowflake schema* menurut Ponniah (2001, p235) lebih kompleks dibandingkan dengan *star schema* karena merupakan pengembangan dari *star schema*, karena tabel-tabel dimensinya merupakan hasil normalisasi dari beberapa tabel yang berhubungan. Prinsip dasar dari skema ini tidak jauh berbeda dari star schema. Dalam menormalisasi tabel dimensi, ada beberapa pilihan yang dapat diperhatikan, antara lain :

a. Secara parsial, lakukan normalisasi hanya beberapa table dimensi saja, dan sisakan yang lain tetap utuh.
b. Secara lengkap atau parsial, lakukan normalisasi hanya pada beberapa tabel dimensi, dan tinggalkan yang tersisa dengan utuh.
c. Secara parsial, lakukan normalisasi pada setiap table dimensi.
d. Secara lengkap, lakukan normalisasi pada setiap table dimensi.

Menurut *Connolly dan Begg* (*2005, p1184*), Skema snowflake merupakan sebuah variasi dari skema bintang dimana tabel dimensi tidak mengandung data denormalisasi. Tabel dimensi diperbolehkan memiliki tabel dimensi lainnya. Berikut adalah contoh gambar snowflake schema. Kelebihan model dimensi snowflake schema **[1]** :

a. Ukuran data lebih kecil di dalam tempat penyimpanan.
b. Lebih mudah dilakukan maintenance dan update.
c. Proses query lebih cepat pada saat proses ETL.

Sedangkan kekurangan modem snowflake schema adalah sebagai berikut :

a. Cenderung lebih sulit dipahami karena kompleksitasnya.
b. Sulit mencari isi karena melihat strukturnya yang kompleks dan bercabang-cabang.



Gambar 2.11 *Snowflake Schema*

3. *Fact Constellation Schema*

   Fact Constellation Schema adalah dimensional model yang didalamnya terdapat lebih dari satu tabel fakta yang membagi satu atau lebih dimension table. Skema ini lebih kompleks daripada star skema karena berisi berbagai tabel fakta. Dalam Fact Constellation Schema, satu dimensi table bisa digunakan di beberapa tabel fakta sehingga desainnya lebih kompleks. Keuntungan dari fact constellation schema adalah kemampuan untuk memodelkan bisnis lebih akurat menggunakan beberapa tabel fakta. Namun kerugiannya adalah sulit dalam pengelolaan dan desain yang rumit.

2.2.4.12 Tujuan Data Warehouse

Menurut Connolly dan Begg (2005, p1152), tujuan *data warehouse* adalah mengintegrasikan data organisasi dengan jumlah yang besar menjadi satu wadah dimana pengguna dapat dengan mudah menjalankan *query*, membuat laporan serta melakukan analisis. Singkatnya, *data warehouse* adalah teknologi manajemen dan analisis data **[1]**.

Data warehouse biasanya diterapkan pada perusanaan yang sudah besar dan memiliki banyak cabang yang mengakibatkan data – data pada perusahaan tersebar dilokasi yang berbeda, sistem operasi yang berbeda, bahkan di basis data yang berbeda. Sehingga dalam pengambilan sebuah keputusan oleh manajemen

perusahaan membutuhkan akses ke semua sumber data pada tiap – tiap cabang dan ini tidak efisien pada waktu. Dari permasalahan yang sedang terjadi ini, *data warehouse* diterapkan sebagai solusi yang kompetitif dan efisien dalam pengambilan keputusan perusahaan. Berikut tujuan diterapkannya perangkat lunak *data warehouse* :

1. Menyediakan data organisasi yang mudah diakses manager
2. Data yang berada di *data warehouse* bersifat konsisten dan merupakan kebenaran.
3. *Data warehouse* merupakan tempat dimana data yang telah digunakan di publikasikan.
4. kualitas data di *data warehouse* dapat diandalkan.

2.3 Alat – alat Pemodelan Sistem

Alat – alat pemodelan sistem membahas mengenai *tools* – *tools* yang digunakan untuk penelitian tugas akhit ini.

2.3.1 Diagram Konteks

Diagram konteks merupakan diagram yang menggambarkan kondisi sistem yang ada baik input maupun output serta menyertakan terminator yang terlibat dalam penggunaan sistem. Diagram ini akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada store dalam diagram konteks.

Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi "siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem." Maka dapat disimpulkan bahwa diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem **[11]**.

2.3.2 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses *fungsional* yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, *model proses*, diagram alur kerja, atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi system.

DFD merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program **[11]**.

2.4 Alat – alat Pembangunan Perangkat Lunak

Dalam penelitian tugas akhir ini ada beberapa alat atau *tools* untuk pembangunan perangkat lunak dalam pembuatan aplikasi *data warehouse.*

2.4.1 C# (C Sharp)

Pada akhir dekade 1990-an, Microsoft membuat program Microsoft Visual J++ sebagai sebuah langkah percobaan untuk menggunakan Java di dalam sistem operasi Windows untuk meningkatkan antarmuka dari Microsoft *Component Object Model* (COM). Akan tetapi, akibat masalah dengan pemegang hak cipta bahasa pemrograman *Java*, *Sun Microsystems*, Microsoft pun menghentikan pengembangan J++, dan beralih untuk membuat pengganti J++, kompilernya dan mesin virtualnya sendiri dengan menggunakan sebuah bahasa pemrograman yang bersifat *general-purpose*. Untuk menangani proyek ini, Microsoft merekrut Anders Helsberg, yang merupakan mantan karyawan Borland yang membuat

bahasa *Turbo Pascal*, dan Borland Delphi, yang juga mendesain *Windows Foundation Classes* (WFC) yang digunakan di dalam J++.

Sebagai hasil dari usaha tersebut, C# pun pertama kali diperkenalkan pada bulan Juli 2000 sebagai sebuah bahasa pemrograman modern berorientasi objek yang menjadi sebuah bahasa pemrograman utama di dalam pengembangan di dalam platform Microsoft .NET Framework. Pengalaman Helsberg sebelumnya dalam pendesain bahasa pemrograman seperti Visual J++, Delphi, Turbo Pascal dengan mudah dilihat dalam sintaksis bahasa C#, begitu pula halnya pada inti *Common Language Runtime* (CLR). Dari kutipan atas interview dan makalah-makalah teknisnya ia menyebutkan kelemahan – kelemahan yang terdapat pada bahasa pemrograman yang umum digunakan saat ini, misalnya C++, Java, Delphi, ataupun Smalltalk. Kelemahan – kelemahan yang dikemukakannya itu yang menjadi basis CLR sebagai bentukan baru yang menutupi kelemahan-kelemahan tersebut, dan pada akhirnya memengaruhi desain pada bahasa C# itu sendiri. Ada kritik yang menyatakan C# sebagai bahasa yang berbagi akar dari bahasa-bahasa pemrograman lain.

Fitur-fitur yang diambilnya dari bahasa C++ dan Java adalah desain berorientasi objek, seperti garbage collection, reflection, akar kelas (root class), dan juga penyederhanaan terhadap pewarisan jamak (multiple inheritance). Fitur-fitur tersebut di dalam C# kini telah diaplikasikan terhadap iterasi, properti, kejadian (event), metadata, dan konversi antara tipe-tipe sederhana dan juga objek.

C# didisain untuk memenuhi kebutuhan akan sintaksis C++ yang lebih ringkas dan *Rapid Application Development* yang 'tanpa batas' (dibandingkan dengan RAD yang 'terbatas' seperti yang terdapat pada Delphi dan Visual Basic). Agar mampu mempromosikan penggunaan besar-besaran dari bahasa C#, Microsoft, dengan dukungan dari Intel Corporation dan Hewlett-Packard, mencoba mengajukan standardisasi terhadap bahasa C#. Akhirnya, pada bulan Desember 2001, standar pertama pun diterima oleh European Computer Manufacturers Association atau Ecma International (ECMA), dengan nomor standar ECMA-334. Pada Desember 2002, standar kedua pun diadopsi oleh

ECMA, dan tiga bulan kemudian diterima oleh International Organization for Standardization (ISO), dengan nomor standar ISO/IEC 23270:2006 **[8]**.

2.4.2 SQL Server

SQL adalah singkatan dari *Structured Query Language*. SQL server adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur client/server. Istilah client, server, dan client/server dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak pada level yang sangat umum **[9]**.

1. *Client*

   Client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (resource) dari komponen sistem lainnya.

2. *Server*

   Server adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

RDBMS adalah dasar untuk SQL, dan untuk semua sistem database modern seperti MS SQL Server, *IBM DB2, Oracle, MySQL, dan Microsoft Access. Data dalam RDBMS disimpan dalam objek database yang disebut tabel.

1. Table

   Table adalah adalah kumpulan data entri terkait dan terdiri dari kolom dan baris.

Structured Query Language (SQL) memiliki beberapa kegunaan dan fungsi SQL itu sendiri diantaranya :

1. SQL memungkinkan anda mengakses dan memanipulasi database.
2. SQL dapat mengeksekusi query terhadap database.
3. SQL dapat mengambil data dari database.
4. SQL dapat menyisipkan catatan dalam database.
5. SQL dapat memperbarui catatan dalam database.
6. SQL dapat menghapus catatan dari database.
7. SQL dapat membuat database baru.

8. SQL dapat membuat tabel baru dalam database.
9. SQL dapat membuat prosedur yang tersimpan dalam database.
10. SQL dapat membuat pemandangan dalam database.
11. SQL dapat mengatur hak akses pada tabel, prosedur, dan pandangan.