

BAB II

LANDASAN TEORI

2.1 Internet

2.1.1 Definisi dan Sejarah Internet

Menurut Deitel & Deitel (2003, p. 12), internet adalah sebuah jaringan komputer yang bersifat global-diperkenalkan di tahun 1960-an dengan dana yang disediakan Departemen Pertahanan Amerika Serikat. Pada awalnya, internet dirancang untuk menghubungkan sistem komputer utama dari beberapa universitas dan organisasi penelitian. Saat ini, internet dapat diakses lebih dari 1 milyar komputer dan peralatan komputer pengendali secara luas.

Dengan pengenalan *World Wide Web* yang memungkinkan pengguna komputer untuk mencari dan melihat dokumen apa saja yang berbasis multimedia melalui internet. Internet telah menjadi salah satu mekanisme komunikasi utama di dunia.

2.1.2 Keuntungan Internet

Menurut Deitel & Deitel (2003, p.12), keuntungan-keuntungan yang bisa didapatkan melalui penggunaan internet antara lain :

- Membuat pekerjaan kita menjadi lebih mudah,
- Membuat informasi secara instant dan mudah diakses secara luas,
- Memungkinkan individu dan bisnis lokal kecil untuk memperkenalkan bisnisnya kepada dunia,
- Mengubah cara suatu bisnis dijalankan,
- Masyarakat luas dapat mencari harga terbaik dari produk atau layanan secara *virtual*,
- Komunitas dengan hobi tertentu dapat saling berhubungan,

- Para peneliti dapat secara cepat dapat menyadari penemuan-penemuan terbaru.

2.1.3 Keamanan Internet

Menurut McLeod (2001, p 77), kelemahan dalam internet yang berpotensi menghambat penggunaan untuk bisnis adalah kurangnya keamanan dalam internet. Pendekatan pertama ialah memisahkan *website* secara fisik dari jaringan internal perusahaan, yang berisi data dan sumber data informasi. Pendekatan kedua adalah dengan memberikan kata sandi (*password*) kepada para mitra dagang (*customer*) yang memungkinkan mereka memasuki jaringan internal melalui internet. Pendekatan ketiga adalah dengan membuat suatu tembok perlindungan yang disebut *firewall*, yaitu suatu filter yang membatasi arus data antar jaringan internal perusahaan dengan internet. Ide dibalik *firewall* adalah dengan membangun satu pengamanan untuk semua komputer di dalam jaringan daripada pengamanan yang terpisah untuk setiap komputer.

2.1.4 World Wide Web (WWW)

WWW adalah suatu sistem server yang besar dan mengatur berbagai macam informasi yang dapat berupa, gambar maupun suara. Internet menyediakan arsitektur jaringan dan WWW menyediakan metode untuk menyimpan dan mengambil dokumen-dokumen (McLeod, 2001, p75). Contoh dari *web browser* yang terkenal antara lain *Internet Explorer*, *Opera*, *Mozilla Firefox* dan *Google Chrome*.

Istilah-istilah dari *World Wide Web* :

1. *Web site*

Web site adalah sebuah komputer yang dikaitkan ke internet yang berisi *hypermedia* yang dapat diakses dari komputer lain melalui suatu *hyperlink*.

2. *Web Page*

Web Page adalah suatu file *hypermedia* yang disimpan di suatu *web site* yang diidentifikasi oleh satu alamat yang unik.

3. *Home Page*

Home Page adalah halaman pertama dari suatu *web site*. Halaman-halaman lain di *site* tersebut dapat dicapai dari *home page*.

4. *Hypertext Link*

Hypertext Link adalah suatu petunjuk yang terdiri dari teks atau grafik yang digunakan untuk mengakses *hypertext* yang disimpan di *web site*. Teks ini biasanya digarisbawahi dan ditampilkan dalam warna biru.

5. *Universal Resource Locator (URL)*

Adalah alamat dari suatu *web page*.

6. *Browser*

Suatu media perangkat lunak yang memungkinkan kita mengambil *hypermedia* dengan mengetikkan parameter pencarian atau mengklik suatu grafik. Kemampuan ini membebaskan kita dari keharusan untuk mengetahui URL dari *web page* yang berisi informasi yang kita butuhkan.

7. *File Transfer Protocol (FTP)*

Adalah suatu perangkat lunak yang memungkinkan kita menyalin file ke komputer kita dari *web site* mana saja.

8. *Transfer Control Protocol / Internet Protocol (TCP/IP)*

Adalah standard komunikasi data yang digunakan oleh komunikasi internet dalam tukar menukar data dari satu komputer ke komputer lain.

9. *Web Server*

Adalah suatu aplikasi pada suatu situs *web* yang bertanggung jawab merespon file dari *web browser*.

10. *Web browser*

Adalah program aplikasi yang digunakan untuk menjelajahi informasi yang ada di halaman *web* dan untuk mendapatkan dokumen dari *web* tersebut.

11. *Hypertext Transfer Protocol (HTTP)*

Adalah suatu protokol yang digunakan agar *client* dapat tetap berhubungan dengan *web server* ataupun sebaliknya.

12. *Hypertext Markup Language (HTML)*

Adalah suatu format data standar yang digunakan untuk membuat dokumen *hypertext* agar dapat dibaca dari suatu platform satu ke platform lainnya, tanpa melakukan perubahan apapun juga.

2.1.5 *e-Commerce*

Menurut Purbo dan Wahyudi (2001, p 36-44), *e-commerce* merupakan satu set teknologi yang dinamis, aplikasi dan proses bisnis yang menghubungkan perusahaan, *customer* dan komunitas tertentu melalui transaksi elektronik, perdagangan barang, pelayanan, dan informasi yang dilakukan secara elektronik.

Menurut O'Brien (2003, p 23), *e-Commerce* adalah proses pembelian dan penjualan serta pemasaran dan pelayanan dari produk, jasa dan informasi melalui jaringan komputer yang beragam.

Menurut Turban et.al (2003, p 275), *e-Commerce* adalah proses jual beli, transfer, atau pertukaran produk, jasa, dan atau informasi melalui jaringan komputer termasuk internet.

Menurut Rayport (2002, p 3), *e-Commerce* digambarkan dengan beberapa atribut, yaitu :

1. Pertukaran informasi digital antara beberapa kelompok (*party*)

Pertukaran informasi ini dapat menggambarkan komunikasi antara 2 kelompok, koordinasi aliran barang-barang dan layanan, atau transmisi pemesanan secara elektronik. Pertukaran ini juga terjadi antara organisasi atau antara individu.

2. Penggunaan teknologi.

e-Commerce bersangkutan dengan transaksi yang menggunakan teknologi. Penggunaan internet *browser* sebagai *interface* dalam *World Wide Web* (WWW), mungkin merupakan penggunaan teknologi yang sudah lazim diketahui oleh para *customer*. Bahkan, *interface* lainnya (seperti ATM, *Electronic Data Interchange-EDI* antara rekan bisnis, dan perbankan secara elektronik melalui telepon) telah merupakan kategori umum dalam *e-Commerce*. Dalam bisnis yang mengatur transaksi secara tetap antara *customer* dan pasar adalah interaksi manusia. Dalam *e-Commerce*, beberapa transaksi dapat diatur menggunakan teknologi.

3. Di dalamnya terdapat aktivitas intra- dan inter- organisasi yang mendukung pertukaran.

Cakupan dari *e-Commerce* termasuk dalam semua aktivitas yang berbasis elektronik baik itu intra- maupun inter- organisasi yang mendukung secara langsung maupun tidak langsung pertukaran dalam transaksi. Dalam hal ini berkaitan erat dengan bagaimana hubungan suatu organisasi bisnis dengan bagian luar dari sistem (seperti *customer*, pemasok, rekan kerja, dan lain-lain) serta bagaimana operasi internal yang dilakukan untuk mengatur aktivitas-aktivitas, proses-proses serta sistem yang ada.

2.1.5.1 Keuntungan *e-Commerce*

Menurut Purbo dan Wahyudi (2001, p 2-3), keuntungan yang dapat diambil dengan adanya *e-Commerce* adalah :

1. *Revenue Stream* (aliran pendapatan) baru yang mungkin lebih menjanjikan, yang tidak dapat ditemui di sistem transaksi tradisional.
2. Dapat meningkatkan *market exposure* (pangsa pasar).
3. Menurunkan *operating cost* (biaya operasional).
4. Melebarkan jangkauan (*Global Reach*).
5. Meningkatkan *customer loyalty*.
6. Meningkatkan *supplier management*.
7. Memperpendek waktu produksi.
8. Meningkatkan *value chain* (mata rantai pendapatan).

2.1.5.2 Resiko dari e-Commerce

Menurut Purbo dan Wahyudi (2001, p 7-9), resiko yang timbul dari penerapan *e-Commerce* adalah :

1. Kehilangan segi finansial secara langsung karena kecurangan.
2. Pencurian informasi rahasia yang berharga.
3. Kehilangan kesempatan bisnis karena gangguan pelayanan.
4. Penggunaan akses ke sumber oleh pihak yang tidak berhak.
5. Kehilangan kepercayaan dari *customer*.
6. Kerugian yang tidak terduga.

2.2 Teori-Teori Basisdata

2.2.1 Pengertian Basisdata

Aplikasi basisdata saat ini merupakan suatu aplikasi yang sudah umum dipakai dalam kehidupan sehari-hari. Sebelum ada aplikasi basisdata seperti DBMS, penyimpanan data masih menggunakan penyimpanan dalam suatu file.

Menurut Connolly & Begg (2005, p15) basisdata adalah suatu kumpulan bersama dari data yang terhubung secara logikal dan deskripsi dari data tersebut, dirancang untuk memenuhi kebutuhan informasi suatu organisasi.

Menurut Gerard V. Post (2005, p2) basisdata adalah koleksi penyimpanan data berdasarkan standar format yang dirancang untuk digunakan bersama – sama oleh *user* yang berbeda – beda.

Menurut Hoffer (2005, p4), basisdata adalah kumpulan data yang terorganisir dan secara logika berkaitan. Terorganisir maksudnya yaitu data distrukturkan sehingga mudah untuk disimpan, dimanipulasi, dan diperoleh oleh *user*. Berkaitan maksudnya data menggambarkan daerah asal (*domain*) kepentingan tertentu bagi kelompok *user*, dan *user* dapat menggunakan data untuk menjawab pertanyaan-pertanyaan seputar domain.

Sedangkan menurut Date (2000, p5), suatu sistem basisdata adalah suatu sistem yang pada dasarnya menyimpan record-record di dalam suatu system yang dilakukan secara komputerisasi, yang tujuannya adalah menjadi suatu kumpulan data yang terhubung dan DBMS (*Database Management Sistem*) menjadi program yang mengatur dan mengontrol akses ke basisdata serta memelihara informasi dan membuat informasi tersebut tersedia berdasarkan peminatan.

Basisdata merepresentasikan *entity*, atribut, dan hubungan relasional antara *entity-entity*. *Entity* merupakan suatu objek nyata (manusia, tempat, benda, konsep, atau kejadian) dalam suatu organisasi yang direpresentasikan dalam basisdata, Atribut merupakan suatu properti yang menjelaskan beberapa aspek dari objek yang ingin disimpan, dan hubungan relasional adalah gabungan *entity-entity* dalam basisdata.

2.2.2 Konsep Basisdata

Menurut Mcleod (2001, p259), dua tujuan konsep basisdata adalah meminimumkan pengulangan data (*data redundancy*) dan mencapai independensi data. Pengulangan data (*data redundancy*) adalah duplikasi data, artinya adalah data yang sama disimpan dalam beberapa *file*. Independensi data

adalah kemampuan untuk membuat perubahan dalam struktur data tanpa membuat perubahan pada program yang memproses data. Independensi data dicapai dengan menempatkan spesifikasi data dalam tabel dan kamus yang terpisah secara fisik dari program. Perubahan pada struktur data hanya dilakukan sekali, yaitu dalam tabel.

2.2.3 Database Management System (DBMS)

Menurut Connolly & Begg (2005, p16), DBMS adalah suatu system perangkat lunak yang memungkinkan *user* untuk mendefinisikan, membuat, memelihara serta mengendalikan akses terhadap basisdata.

Menurut Hoffer (2005, p7), DBMS adalah sistem yang digunakan untuk membuat, merawat, dan menyediakan kontrol akses untuk *user* basisdata. DBMS menyediakan metode sistematis untuk pembuatan, *updating*, penyimpanan dan penerimaan data pada basisdata. DBMS juga menyediakan fasilitas untuk mengontrol akses data, menguatkan integritas data, mengatur kontrol konkurensi, dan menyimpan basisdata.

Sedangkan menurut Kadir (2000, p17) DBMS diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi, dan memperoleh data informasi dengan praktis dan efisien.

Menurut Petroustos (2000, p5), DBMS adalah kumpulan program yang digunakan untuk membuat dan mengelola basis data. DBMS menyediakan fungsi-fungsi sebagai berikut :

1. DBMS mengizinkan aplikasi mendefinisikan struktur dari basis data dengan pernyataan SQL. Pernyataan SQL yang mendefinisikan atau mengedit struktur ini disebut dengan *Data Definition Language (DDL)*.
2. DBMS mengizinkan aplikasi memanipulasi informasi yang disimpan didalam basis data dengan pernyataan SQL. Pernyataan SQL yang memanipulasi informasi ini disebut dengan *Data Manipulation Language (DML)*.

3. DBMS melindungi integritas basis data dengan menerapkan beberapa aturan, yang dimasukkan kedalam perancangan basis data tersebut.

Beberapa komponen DBMS (Connolly & Begg, 2005, p18) :

1. *Hardware* (perangkat keras)

Hardware dibutuhkan untuk menjalankan DBMS dan aplikasiaplikasi. *Hardware* dapat berkisar dari komputer pribadi tunggal, ke suatu mainframe tunggal, ke suatu jaringan komputer. *Hardware* yang khusus tergantung pada kebutuhan perusahaan dan DBMS yang digunakan.

2. *Software* (perangkat lunak)

Komponen *software* terdiri dari *software* DBMS itu sendiri dan program aplikasi, bersama dengan sistem operasi, termasuk *software* jaringan jika DBMS digunakan melalui jaringan. Khusus program aplikasi ditulis menggunakan *third-generation programming language* (3GL), seperti C, C++, Java, Visual Basic, COBOL, Fortran, Ada atau Pascal, atau menggunakan *fourth-generation programming language* (4GL) seperti SQL, yang disertakan dalam *third-generation language*.

3. Data

Data merupakan komponen paling penting dari lingkungan DBMS, khususnya dari sudut pandang *end-users*. Pada umumnya data dalam sebuah sistem basisdata baik itu *single-user system* maupun *multi-user system* bersifat *integrated* dan *shared*.

4. Prosedur

Prosedur merupakan instruksi dan aturan yang mengendalikan perancangan dan penggunaan basisdata. Instruksi yang ada digunakan untuk menjalankan sistem, seperti :

- Memasuki DBMS.
- Menggunakan suatu fasilitas DBMS khusus atau program aplikasi.
- Menjalankan dan menghentikan DBMS.
- Membuat duplikat backup basisdata.
- Menangani kesalahan-kesalahan *hardware* atau *software*.

- Merubah struktur dari suatu tabel, mengatur ulang basisdata melalui *multiple disks*, meningkatkan kinerja, atau menyimpan data ke *secondary storage*.

5. Manusia

Komponen terakhir adalah manusia yang terlibat dengan sistem. Terdiri dari *data administrators*, *database administrators*, *database designers*, *application developers*, dan *end-users*.

Menurut Connolly & Begg (2005, p26), DBMS mempunyai keuntungan yang potensial, meskipun sayangnya juga mempunyai kerugian. Adapun keuntungan menggunakan DBMS antara lain adalah sebagai berikut :

1. Mengontrol redundansi data

Maksudnya adalah menghilangkan redundansi dengan mengintegrasikan file-file sehingga banyak salinan data yang sama tidak disimpan.

2. Konsistensi data

Dengan menghilangkan atau mengontrol redundansi data, maka dapat dikurangi resiko terjadinya ketidak konsistenan.

3. Informasi yang lebih banyak dari sejumlah data yang sama

Dengan pengintegrasian data operasional, memungkinkan organisasi untuk memperoleh informasi tambahan dari data yang sama.

4. Berbagi data

Data dapat dibagi ke seluruh organisasi oleh *user* yang berotorisasi.

5. Meningkatkan integrasi data

Integrasi basisdata merujuk kepada validitas dan kekonsistenan data yang tersimpan.

6. Meningkatkan keamanan

Melindungi basisdata dari *user* yang tidak berotorisasi.

7. Menerapkan standar

Integrasi memungkinkan DBA untuk mendefinisikan dan menerapkan standar yang penting.

8. Skala ekonomi

Menggabungkan data operasional organisasi dan menciptakan sekumpulan aplikasi yang bekerja pada satu sumber data dapat menciptakan penghematan biaya.

9. Mengatasi kebutuhan *user* yang kompleks

Tiap *user* mempunyai kebutuhan yang berbeda-beda satu sama lain, sehingga basisdata perlu dikontrol oleh DBA yang bertugas untuk mengambil keputusan mengenai perancangan dan pengoperasian basisdata.

10. Meningkatkan aksesibilitas data dan respon

Data yang melintasi batas departemen dapat secara langsung diakses oleh *user* akhir.

11. Meningkatkan produktivitas

Sebuah DBMS menyediakan banyak fungsi-fungsi standar dimana seorang *programmer* dapat membuatnya di dalam aplikasi berbasis file. Hal ini dapat meningkatkan produktivitas dari seorang *programmer* dan mengurangi waktu pengembangan.

12. Meningkatkan pemeliharaan melalui data independen

Di dalam sebuah sistem berbasis file, deskripsi dari data dibangun ke dalam masing-masing aplikasi sehingga membuat program dependen terhadap data. Sedangkan dengan menggunakan DBMS, deskripsi dari data dipisahkan dari program aplikasinya sehingga membuat aplikasi tahan terhadap perubahan dari deskripsi data. Hal ini disebut dengan data yang independen.

13. Meningkatkan konkurensi

Mencegah terganggunya integritas data karena dua atau lebih *user* dapat mengakses data yang sama.

14. Mengembangkan *backup* dan pemulihan

Memudahkan *user* untuk mem-*backup* data dan memulihkan data yang rusak. Disamping keuntungan-keuntungan tersebut, DBMS juga mempunyai kerugian antara lain sebagai berikut :

1. Kompleksitas

User harus benar-benar memahami fungsi-fungsi untuk dapat menggunakannya secara penuh.

2. Ukuran

Kompleksitas dari fungsionalitas membuat DBMS menjadi sebuah perangkat lunak dengan ukuran yang besar dan membutuhkan memori yang cukup besar untuk dapat berjalan secara efisien.

3. Biaya DBMS

Biaya secara signifikan beragam tergantung pada lingkungan dan fungsionalitas.

4. Biaya perangkat keras tambahan

Kebutuhan penyimpanan untuk DBMS dan basisdata mungkin membutuhkan pembelian tambahan media penyimpanan.

5. Biaya konversi

Dibutuhkan biaya pelatihan dan perekrutan tenaga ahli dalam konversi dan pelaksanaan sistem.

6. Kinerja

Aplikasi tidak dapat berjalan secepat biasanya karena DBMS ditulis menjadi lebih umum untuk banyak aplikasi.

7. Dampak kegagalan yang lebih tinggi

Pemusatan sumber meningkatkan kelemahan sistem.

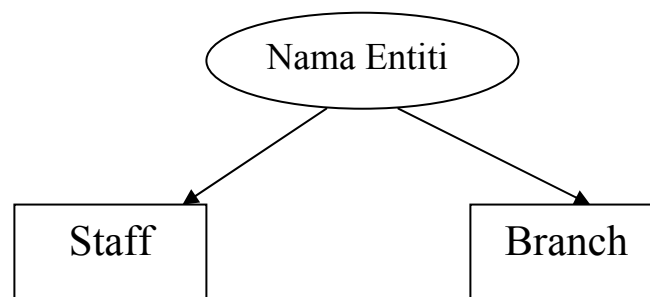
2.2.4 Entity – Relationship Modeling

Menurut Connolly (2002, p330), *Entity – Relationship Modeling* adalah pendekatan perancangan basis data *top down* yang dimulai dengan mengidentifikasi data yang penting yang disebut dengan entiti dan hubungan di antara data yang harus direpresentasikan dalam model.

Dalam merancang basis data, hal yang paling penting adalah dengan menggunakan *Entity Relationship* (ER). Karena tanpa ER, bisa dipastikan proses pembuatan basis data berjalan lama dan tidak teratur. Selain itu yang perlu diperhatikan adalah membuat relasi – relasi yang benar di antara tabel. Proses perancangan basis data memerlukan waktu yang lama jika basis datanya besar. Pendokumentasian rancangan basis data mutlak harus dilakukan dengan baik, agar mudah didalam pengembangan dan perbaikan nantinya.

2.2.4.1 Entity (Entiti)

Konsep dasar dari Model ER adalah Entiti, yaitu kumpulan dari objek -objek dari dunia nyata dengan sifat (properti) yang sama, yang diidentifikasi oleh perusahaan yang keberadaanya tidak tergantung.



Gambar 2.1 Contoh Tipe Entiti(Connolly, 2002, p333)

2.2.4.2 Attribute (Atribut)

Merupakan sifat-sifat dari sebuah entiti atau tipe relasi. Contoh : sebuah entiti Staff digambarkan oleh atribut staffNo, nama, jabatan, dan gaji. Atribut domain adalah himpunan nilai yang diperbolehkan untuk satu atau lebih atribut. Macam-macam atribut :

- *Simple Attribute* yaitu atribut yang terdiri dari satu komponen tunggal dengan keberadaan yang independen dan tidak dapat dibagi menjadi bagian yang lebih kecil lagi. Dikenal juga dengan nama *Atomic Attribute*.
- *Composite Attribute* yaitu atribut yang terdiri dari beberapa komponen, dimana masing - masing komponen memiliki keberadaan yang

independen. Misalkan atribut alamat dapat terdiri dari nama jalan, kota, kode pos.

- *Single-valued Attribute* yaitu atribut yang mempunyai nilai tunggal untuk setiap kejadian. Misalnya entiti Cabang memiliki satu nilai untuk atribut branchNo pada setiap kejadian.
- *Multi-valued Attribute* yaitu atribut yang mempunyai beberapa nilai untuk setiap kejadian. Misal entiti Cabang memiliki beberapa nilai untuk atribut telpNo pada setiap kejadian.
- *Derived Attribute* yaitu atribut yang memiliki nilai yang dihasilkan dari satu atau beberapa atribut lainnya, dan tidak harus berasal dari satu entiti.

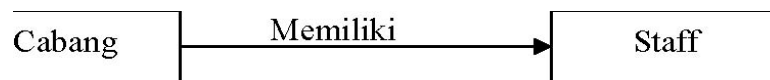
2.2.4.3 Keys

Candidate key didefinisikan sebagai sejumlah minimal atribut – atribut yang nilainya dapat secara unik mengidentifikasi suatu entiti. (Connolly, 2002, p340).

Primary key didefinisikan sebagai *candidate key* yang terpilih guna secara unik mengidentifikasi suatu entiti. (Connolly, 2002, p341).
Composite key didefinisikan sebagai *candidate key* yang terdiri atas 2 atau lebih atribut. (Connolly, 2002, p341).

2.2.5 Relationship (relasi)

Relasi didefinisikan sebagai hubungan yang terjadi antar entiti. Setiap hubungan diberi nama sesuai dengan fungsinya. Representasi diagram dari relasi adalah berupa garis lurus yang menghubungkan 2 (dua) buah entiti.

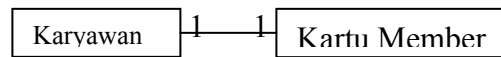


Gambar 2.2 Contoh Relasi (Connolly, 2002, p335)

Jenis-jenis basis data relasional atau hubungan yang bisa terjadi antara satu entiti dengan entiti lain dalam sebuah basis data, meliputi :

1. *One-to-One* (1 : 1)

Contohnya hubungan yang terjadi antara entiti “Karyawan” dengan “Kartu Member” :

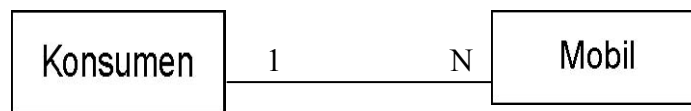


Gambar 2.3 Relasi *One to One*

Seorang karyawan hanya diizinkan memiliki satu kartu member, dan satu kartu member hanya bisa dimiliki oleh satu orang karyawan.

2. *One-to-Many* (1 : *)

Contohnya hubungan yang terjadi antara entiti “Konsumen” dengan “Mobil” :

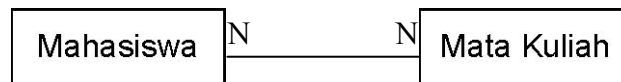


Gambar 2.4 Relasi *One to Many*

Seorang konsumen bisa membeli lebih dari satu mobil, tetapi satu mobil tidak dapat diatasmamakan lebih dari satu orang konsumen.

3. *Many-to-Many* (* : *)

Contohnya hubungan yang terjadi antara entiti “Mahasiswa” dengan “Mata Kuliah” :



Gambar 2.5 Relasi *Many to Many*

Seorang mahasiswa bisa mengikuti lebih dari satu mata kuliah, dan satu matakuliah bisa diikuti lebih dari satu mahasiswa.

2.2.6 Normalisasi

Menurut Whitten (2001, p288) normalisasi adalah teknik analisis data yang mengatur atribut - atribut data dengan mengelompokkannya dalam bentuk yang tidak berulang, fleksibel, dan disesuaikan dengan basis data.

Menurut Hoffer (2002, p189) normalisasi adalah proses mendekomposisi relasi yang mengandung anomali guna menghasilkan relasi yang lebih sederhana dan terstruktur dengan baik.

Menurut Connolly (2002, p379) normalisasi adalah suatu teknik untuk mengorganisasikan data ke dalam tabel – tabel, guna memenuhi kebutuhan data dalam sebuah organisasi. Tujuan dari normalisasi adalah untuk menghilangkan kerangkapan data, untuk mengurangi kompleksitas dan untuk mempermudah modifikasi data.

2.2.6.1 Bentuk – bentuk Normalisasi

1. Unnormalized Form (UNF)

Menurut Connolly (2002, p387), UNF merupakan suatu tabel yang berisikan satu atau lebih group yang berulang.

2. First Normal Form (1NF)

Menurut Connolly (2002, p388), 1NF adalah relasi dimana pertemuan antar setiap baris dan kolom terdiri dari 1 (satu) dan hanya 1 (satu) nilai. Dalam normalisasi ini data yang berulang – ulang dihilangkan.

3. Second Normal Form (2NF)

Menurut Connolly (2002, p392), 2NF adalah sebuah relasi dalam 1NF yang setiap atribut bukan *primary key* (PK) bersifat *Full Functional Depedency* pada *primary key* dari relasi tersebut.

- *Full Functional Depedency* (FFD)

Menurut Connolly (2002, p392), *Full Function Depedency* adalah mengindikasikan kondisi dimana jika A dan B adalah atribut pada suatu relasi, B dianggap tergantung fungsional penuh terhadap A, jika B

tergantung fungsional terhadap A, tetapi bukan merupakan himpunan bagian A.

4. Third Normal Form (3NF)

Menurut Connolly (2002, p394), 3NF adalah sebuah relasi yang memenuhi 1NF dan juga 2NF dimana tidak terdapat atribut bukan *primary key* yang bersifat *Transitive Dependency*.

- *Transitive Dependency*

Menurut Connolly (2002, p394), *Transitive Dependency* adalah kondisi dimana A, B, C merupakan atribut – atribut dari suatu relasi sedemikian sehingga $A \rightarrow B$ dan $B \rightarrow C$, maka C *Transitive Dependency* terhadap A melalui B (dengan syarat A bukan *Transitive Dependency* dari B atau C).

5. Boyce – Codd Normal Form (BCNF)

Menurut Connolly (2002, p412), BCNF adalah sebuah relasi dimana setiap penentu atau determinan adalah *candidatekey*.

6. Fourth Normal Form (4NF)

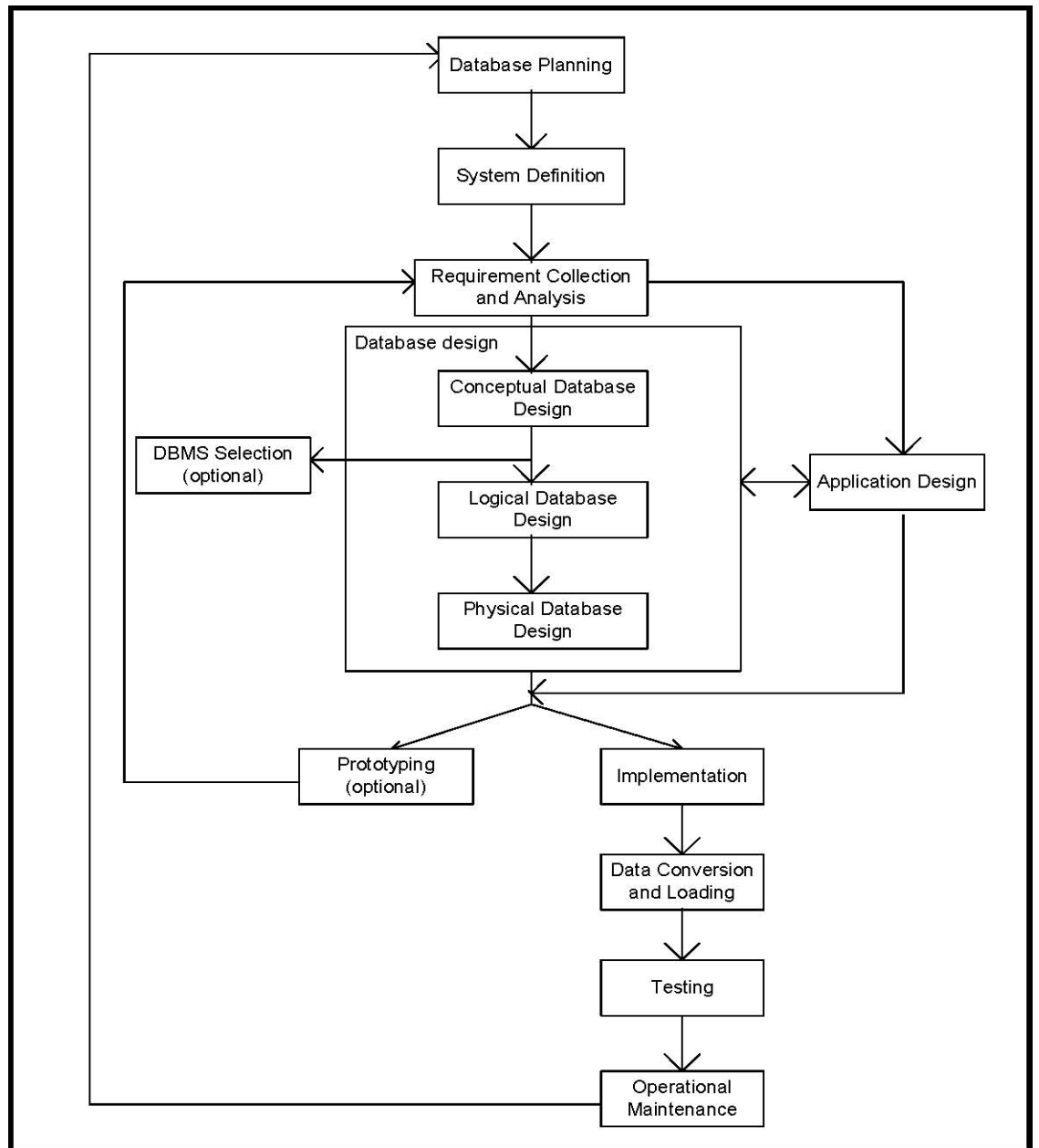
Normalisasi ke 4 (empat) dilakukan untuk menghilangkan *multi valued dependency*.

7. Fifth Normal Form (5NF)

Normalisasi ke 5 (lima) menyebabkan relasi tidak mempunyai *join dependency*.

2.2.7 Siklus Hidup Aplikasi Basis Data

Tahapan siklus hidup aplikasi Basis data yang tampak pada gambar 2.6 berikut ini tidak mutlak dilaksanakan secara terurut, melainkan melalui sejumlah pengulangan dari tahapan terdahulu agar didapatkan hasil semaksimal mungkin.



Gambar 2.6 Tahapan dalam daur hidup aplikasi Basis Data (Connolly, 2002, p272)

Penjelasan dari tahapan – tahapan pada gambar 2.6 adalah sebagai berikut :

2.2.7.1 Database Planning

Merupakan aktivitas manajemen yang memungkinkan tahapan dari database system development lifecycle direalisasikan seefektif dan seefisien mungkin. Perencanaan basisdata harus terintegrasi dengan

keseluruhan strategi sistem informasi dari organisasi. Terdapat 3 hal pokok yang berkaitan dengan strategi sistem informasi, yaitu :

- Identifikasi rencana dan sasaran dari perusahaan termasuk mengenai sistem informasi yang dibutuhkan.
- Evaluasi sistem informasi yang ada untuk menetapkan kelebihan dan kekurangan yang dimiliki.
- Penaksiran kesempatan teknologi informasi yang mungkin

Metodologi untuk mengatasi hal tersebut diatas, yaitu :

- Database Planning - Mission Statement Mission statement untuk proyek basisdata mendefinisikan tujuan utama dari aplikasi basisdata. Mengarahkan proyek basisdata, biasanya mendefinisikan mission statement. Mission statement membantu menjelaskan kegunaan dari proyek basisdata dan menyediakan alur yang lebih jelas untuk mencapai efektifitas dan efisiensi penciptaan dari suatu aplikasi basisdata yang diinginkan.

- Database Planning - Mission Objectives
Ketika mission statement telah didefinisikan, maka mission objectives didefinisikan. Setiap tujuan harus mengidentifikasi tugas khusus yang harus didukung oleh basisdata. Dapat juga disertai dengan beberapa informasi tambahan yang menspesifikasikan pekerjaan yang harus diselesaikan, sumber data yang digunakan dan biaya untuk membayar kesemuanya itu.

Database planning juga harus menyertakan pengembangan standar-standar yang menentukan:

- Bagaimana data akan dikumpulkan,
- Bagaimana menspesifikasikan format / bentuk data,
- Dokumentasi penting apakah yang akan diperlukan,
- Bagaimana desain dan implementasi harus dilakukan.

2.2.7.2 System Definition

Menjelaskan batasan-batasan dan cakupan dari aplikasi basisdata dan sudut pandang user (user view) yang utama. User view mendefinisikan apa yang diwajibkan dari suatu aplikasi basisdata dari perspektif aturan kerja khusus atau area aplikasi perusahaan. Aplikasi basisdata dapat memiliki satu arah atau lebih User view. Identifikasi User view membantu memastikan bahwa tidak ada kebutuhan yang terlupakan ketika pembuatan aplikasi baru yang dibutuhkan. User view juga membantu dalam pengembangan aplikasi basisdata yang kompleks yang memungkinkan permintaan-permintaan dipecah ke dalam bagian-bagian yang lebih sederhana.

2.2.7.3 Requirement Collection and Analysis

Suatu proses pengumpulan data dan analisa informasi mengenai bagian organisasi yang didukung oleh aplikasi basisdata, dan menggunakan informasi tersebut untuk identifikasi kebutuhan user akan sistem yang baru. Informasi yang dikumpulkan untuk setiap user view utama meliputi :

- Deskripsi data yang digunakan atau dihasilkan.
- Detil mengenai bagaimana data digunakan atau dihasilkan.
- Beberapa kebutuhan tambahan untuk aplikasi basisdata baru.

Informasi dianalisa untuk identifikasi kebutuhan agar disertakan dalam alokasi basisdata yang baru. Aktivitas penting lainnya adalah menentukan bagaimana mengatur aplikasi basisdata dengan multiple user view, yaitu :

- Pendekatan Terpusat (Centralized approach)

Kebutuhan untuk setiap user view digabungkan menjadi sekumpulan kebutuhan. Sebuah global data model dibuat berdasarkan atas penggabungan kebutuhan.

- Pendekatan Integrasi View (View integration approach)

Kebutuhan untuk setiap user view digunakan untuk membangun model data terpisah untuk merepresentasikan user view tersebut. Hasil dari model data tersebut nantinya akan digabungkan dalam tahapan desain basisdata. Model-model yang merepresentasikan user view tunggal disebut local data model, dan tersusun atas diagram-diagram dan dokumentasi yang secara formal menggambarkan kebutuhan dari user view khusus terhadap basisdata. Kemudian local data model digabungkan untuk menghasilkan global data model yang merepresentasikan seluruh view user untuk basisdata.

- Kombinasi keduanya (Combination of both approaches).

2.2.7.4 Database Design

Merupakan suatu proses pembuatan sebuah desain basisdata yang akan mendukung tujuan dan operasi suatu perusahaan. Pada tahap ini dilakukan 3 tahap perancangan yakni:

- Perancangan konseptual, dilakukan untuk membuat representasi konseptual dari basisdata, termasuk identifikasi *entity-entity* yang penting beserta atribut-atribut dan relasi-relasinya.
- Perancangan logikal, dilakukan untuk menterjemahkan representasi konseptual ke dalam struktur logikal dari basisdata, termasuk desain relasi-relasinya.
- Perancangan fisikal, dilakukan untuk menentukan bagaimana struktur logikal diimplementasikan secara fisik ke dalam target DBMS (*Database Management System*).

Tujuan utamanya adalah :

- Merepresentasikan data dan *relationship* antar data yang dibutuhkan oleh seluruh area aplikasi utama dan *user group*.

- Menyediakan model data yang mendukung segala transaksi yang diperlukan pada data.
- Menspesifikasikan desain minimal yang secara tepat disusun untuk memenuhi kebutuhan kinerja yang ditetapkan pada sistem.

Pendekatan dalam desain basisdata :

1. *Top-down*

Diawali dengan pembentukan model data yang berisi beberapa *entity* tingkat tinggi dan relationship, yang kemudian menggunakan pendekatan *top-down* secara berturut-turut untuk mengidentifikasi *entity* dengan tingkat yang lebih rendah, hubungan relasional dan atribut lainnya.

2. *Bottom-up*

Dimulai dari atribut dasar, dengan analisis dari penggabungan antar atribut yang dikelompokkan ke dalam suatu relasi yang merepresentasikan tipe dari *entity* dan hubungan relasional antar *entity*.

3. *Inside-out*

Berhubungan dengan pendekatan *bottom-up* tetapi sedikit berbeda dengan identifikasi awal *entity* utama dan kemudian menyebar ke *entity*, hubungan relasional, dan atribut terkait lainnya yang lebih dulu diidentifikasi.

4. *Mixed*

Menggunakan pendekatan *bottom-up* dan *top-down* untuk bagian yang berbeda sebelum pada akhirnya digabungkan

2.2.7.5 Database Management System selection (optional)

Tahap ini melakukan pemilihan DBMS yang sesuai guna mendukung aplikasi basis data (Connolly, 2002, p284). Pemilihan dapat dilakukan dengan beberapa langkah berikut ini :

- Definisikan persyaratan atas referensi pemilihan DBMS.
- Buat daftar beberapa produk yang dapat dijadikan pilihan.
- Evaluasi fitur dari masing – masing produk.
- Buat rekomendasi pemilihan beserta laporannya.

Pemilihan DBMS juga bisa dilakukan dengan berdasar pada Analisis Kelayakan (*Feasibility Analysis*) (Whitten, 2001, p404). Analisis Kelayakan ini terbagi atas 4 hal, yaitu :

- Kelayakan Operasional (*Operational Feasibility*)

Kelayakan operasional merupakan ukuran sejauh mana solusi dapat berguna bagi perusahaan dan juga mengukur bagaimana penerimaan orang – orang terhadap solusi itu. Kelayakan operasional berorientasi pada user.

- Kelayakan Teknis (*Technical Feasibility*)

Kelayakan teknis mengukur seberapa praktis solusi teknis dan ketersediaan sumber daya teknis keahlian yang dibutuhkan. Kelayakan teknis berorientasi pada komputer.

- Kelayakan Jadwal (*Schedule Feasibility*)

Kelayakan jadwal merupakan ukuran yang menyatakan seberapa masuk akal tengat waktu proyek dan menentukan apakah waktu yang dialokasikan untuk sebuah proyek akurat.

- Kelayakan Ekonomi (*Economic Feasibility*)

Kelayakan ekonomi mengukur efektifitas biaya dari suatu proyek atau solusi. Kelayakan ekonomi berhubungan dengan biaya dan manfaat

yang diperoleh dari Sistem Informasi. Garis dasar kebanyakan proyek adalah kelayakan ekonomi.

2.2.7.6 Application Design

Merupakan suatu proses perancangan *user interface* dan program aplikasi yang menggunakan dan memproses basisdata. Ada dua aktifitas penting dalam perancangan aplikasi yaitu :

1. *Transaction Design* (Perancangan Transaksi)

Transaksi merupakan sebuah aksi, atau serangkaian aksi yang dilakukan oleh seorang *user* atau program aplikasi yang mengakses atau mengubah isi dari basisdata. Kegunaan dari desain transaksi adalah untuk mendefinisikan dan mendokumentasikan karakteristik *high level* dari suatu transaksi yang dibutuhkan pada basisdata, meliputi :

- data yang akan digunakan oleh transaksi
- karakteristik fungsional dari suatu transaksi
- *output* transaksi
- keuntungan bagi *user*
- tingkat kegunaan yang diharapkan.

Ada 3 jenis transaksi yaitu :

- *Retrieval transaction* yaitu transaksi yang digunakan untuk mengambil data untuk informasi di layar atau dalam sebuah laporan. Contoh : operasi untuk menampilkan detail dari sebuah barang yang telah diberikan kode barangnya
- *Update transaction* yaitu transaksi yang digunakan untuk memasukkan data baru, menghapus data lama, atau merubah data yang sudah ada di dalam sebuah database. Contoh : operasi untuk memasukkan detail dari sebuah barang baru ke dalam database.

- *Mixed Transactions* yaitu transaksi yang merupakan perpaduan dari retrieval dan update transactions. Contoh : operasi untuk menampilkan detail dari suatu barang dengan menggunakan kode barangnya, dan kemudian merubah detail dari barang tersebut (mis: harga).

2. *User Interface Design* (Perancangan antarmuka)

Merupakan rancangan *layout* (tampilan) yang dibuat terlebih dahulu sebelum mengimplementasikan suatu *form* atau laporan.

Beberapa aturan pokok dalam pembuatan *user interface*:

- *Meaningful title*, diusahakan pemberian nama suatu *form* cukup jelas dan tidak ambigu dalam mengidentifikasi kegunaan dari suatu *form / report*.
- *Comprehensible instructions*, penggunaan terminologi yang familiar untuk menyampaikan instruksi ke *user* dan jika informasi tambahan diperlukan, maka harus disediakan *helpscreen*.
- *Logical grouping and sequencing of fields*, *fields* yang saling berhubungan ditempatkan pada *form / report* yang sama. Urutan *field* harus logis dan konsisten.
- *Visually appealing layout of the form / report*, tampilan *form / report* harus menarik, dan sesuai dengan *hardcopy* agar konsisten.
- *Familiar field labels*, penggunaan *field* yang familiar
- *Consistent terminology and abbreviation*, terminologi dan singkatan yang digunakan harus konsisten.
- *Consistent use of color*.
- *Visible space and boundaries for data-entry fields*, jumlah tempat yang disediakan untuk *data entry* harus diketahui oleh *user*.

- *Convinient cursor movement*, *user* dapat dengan mudah menjalankan operasi yang diinginkan dengan menggerakkan kursor pada *form / report*.
- *Error correction for individual characters and entire fields*, *user* dapat dengan mudah menjalankan operasi yang diinginkan dan melakukan perubahan terhadap nilai *fields*.
- *Error messages for unacceptable values*.
- *Optional fields market clearly*.
- *Explanatory messages for fields*, ketika *user* meletakkan kursor pada suatu *field*, maka keterangan mengenai *field* tersebut harus dapat dilihat.
- *Completion signal*, indikator yang menjelaskan bahwa suatu proses telah selesai dilaksanakan.

2.2.7.7 Prototyping (optional)

Pada tahap ini membangun suatu model kerja dari aplikasi basis data. Suatu *prototype* tidak selalu harus memiliki seluruh fitur – fitur yang dibutuhkan atau menyediakan fungsi menyeluruh dari hasil akhir sistem. Tujuan utama mengembangkan suatu *prototype* aplikasi basis data adalah memberikan kesempatan kepada *user* untuk menggunakan *prototype* guna mengidentifikasi fitur - fitur apakah yang sudah bekerja dengan baik, ataupun masih memiliki kekurangan, agar dapat dilakukan perbaikan terhadap aplikasi basis data, dan jika mungkin meningkatkan fitur baru kepada aplikasi basis data. Dengan cara ini kita dapat memperjelas kebutuhan pemakai dan pengembang sistem dalam mengevaluasi kelayakan dengan sistem tertentu. (Connolly, 2002, pp291 - 292).

Ada 2 (dua) strategi prototyping yang umum digunakan, yaitu :

- *Requirement Prototyping*

Menggunakan *prototype* untuk menetapkan kebutuhan dari tujuan aplikasi basis data dan ketika kebutuhan sudah terpenuhi, *prototype* tidak digunakan lagi atau dibuang.

- *Evolutionary Prototyping*

Digunakan dengan tujuan yang sama, tetapi perbedaannya adalah *prototype* tetap digunakan.

2.2.7.8 Implementation

Pada tahap ini membangun realisasi fisik dari basis data dan rancangan aplikasinya. Implementasi dari basis data dilakukan dengan membangun struktur dan file dari basis data yang juga berkaitan dengan penggunaan DBMS tertentu. Sementara program aplikasinya diimplementasikan dengan menggunakan bahasa pemrograman yang telah ditentukan. (Connolly, 2002, p292).

2.2.7.9 Data Conversion and Loading

Pada tahap ini merupakan pemindahan data yang ada ke dalam basis data yang baru dan melakukan konversi terhadap aplikasi yang ada untuk digunakan pada basis data yang baru tersebut. (Connolly, 2002, p292).

Tahapan ini diperlukan hanya ketika suatu sistem basis data baru menggantikan sistem basis data yang lama. DBMS biasanya mempunyai suatu kegunaan yang memanggil *file* yang sudah ada ke dalam basis data yang baru. Dapat juga mengubah dan menggunakan program aplikasi dari sistem yang lama untuk digunakan oleh sistem yang baru.

2.2.7.10 Testing

Tahap pengeksekusian program aplikasi dengan tujuan menemukan *error* (kesalahan). Aplikasi basis data diuji dan divalidasi terhadap kebutuhan yang telah dispesifikasi oleh *user*. (Connolly, 2002, p293)

2.2.7.11 Operational Maintenance

Tahap ini merupakan suatu proses pengawasan dan pemeliharaan sistem setelah instalasi. Sistem diawasi dan dipelihara secara berkesinambungan. Dan jika dibutuhkan, fitur – fitur baru akan *upgrade*(ditambahkan). (Connolly, 2002, p293).

2.2.8 Perancangan Basis Data

Menurut Connolly (2002, p 419), perancangan basis data adalah proses pembuatan sebuah rancangan untuk sebuah basis data yang mendukung operasi dan tujuan dari perusahaan. Perancangan basis data dibagi ke dalam 3 (tiga) tahapan utama, yaitu Perancangan basis data konseptual, Perancangan basis data logikal, dan Perancangan basis data fisikal.

2.2.8.1 Perancangan Basis Data Konseptual

Proses membangun sebuah model data dari informasi yang diperoleh dari suatu perusahaan yang bebas dari pertimbangan fisikal. (Connolly, 2002, p419).

Berikut ini adalah langkah-langkah dalam Perancangan Basis Data Konseptual :

1. Membangun model data konseptual lokal untuk setiap *view*
2. Mengidentifikasi tipe entiti.

3. Mengidentifikasi tipe relasi.
4. Mengidentifikasi dan menghubungkan atribut dengan tipe entiti atau relasi.
5. Menentukan domain atribut.
6. Menentukan atribut *candidate key* dan *primary key*.
7. Mempertimbangkan penggunaan *enhanced modeling concepts* (optional).
8. Memeriksa model terhadap redundansi.
9. Memvalidasi model data konseptual lokal terhadap transaksi pengguna.
10. Meninjau ulang model data konseptual lokal dengan pengguna.

2.2.8.2 Perancangan Basis Data Logikal

Proses untuk membuat sebuah rancangan informasi yang digunakan dalam perusahaan berdasarkan suatu model data spesifik, tetapi masih bebas dari DBMS dan pertimbangan – pertimbangan fisik lainnya. (Connolly, 2002, p419).

Berikut ini adalah langkah – langkah dalam Perancangan Basis Data Logikal :

1. Membuat dan memvalidasi model data logikal lokal untuk setiap *view*.
2. Menghilangkan fitur – fitur yang tidak kompatibel dengan model relasional (optional).
3. Membuat hubungan relasi untuk model data logikal lokal.
4. Memvalidasi hubungan dengan menggunakan normalisasi.
5. Validasi relasi hubungan terhadap transaksi pengguna.

6. Menentukan *integrity constraints*.
7. Meninjau ulang model data logikal lokal dengan pengguna.
8. Membangun dan memvalidasi model data logikal global.
9. menggabungkan model data logikal lokal ke dalam model global.
10. Validasi model data logikal global.
11. Memeriksa perkembangan di mana yang akan datang.
12. Meninjau kembali model data logikal global dengan pengguna.

2.2.8.3 Perancangan Basis Data Fisikal

Proses menghasilkan sebuah deskripsi dan implementasi basis data pada media penyimpanan sekunder, dengan menggambarkan hubungan dasar, organisasi *file* dan indeks yang digunakan untuk memperoleh akses yang efisien terhadap data, beserta segala *integrity constraints* yang terkait dan pertimbangan keamanan. (Connolly, 2002, p419).

Berikut ini adalah langkah – langkah dalam Perancangan Basis Data Fisikal :

1. Menerjemahkan model data logikal global untuk DBMS.
2. Merancang relasi dasar.
3. Merancang gambaran dari data yang telah didapat.
4. Merancang *enterprise constraints*.
5. Merancang representasi fisik.
6. Menganalisa transaksi.
7. Memilih file organisasi.
8. Memilih indeks.

9. Memperkirakan kebutuhan *disk space*.
10. Merancang *user views*.
11. Mendesain mekanisme keamanan.
12. Mempertimbangkan penggunaan dari redundansi terkontrol.
13. Melakukan pengawasan dan pemeliharaan terhadap sistem operasional

2.2.8.4 Bagan alir dokumen

Menurut Mulyadi (1993, p60), bagan alir dokumen adalah simbol – simbol standar yang digunakan oleh bagian analisis sistem untuk membuat bagan alir dokumen yang menggambarkan sistem tertentu. Sebenarnya banyak cara untuk mengembangkan bagan alir dokumen suatu sistem, namun bagan alir dokumen merupakan alat yang digunakan secara luas oleh para penulis untuk menggambarkan aliran dokumen suatu sistem.

Adapun simbol yang digunakan dalam bagan alir dokumen yaitu :

1. Simbol proses.

Menunjukkan kegiatan proses dari operasi program komputer.



2. Simbol dokumen

Menunjukkan dokumen input dan output, baik untuk proses manual, mekanik, maupun komputer.



3. Simbol kegiatan manual

Menunjukkan kegiatan manual.



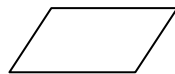
4. Simbol penghubung

Menunjukkan penghubung ke halaman yang sama dan ke halaman yang lain.



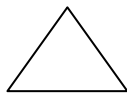
5. Simbol catatan

Menunjukkan catatan data yang direkam sebelumnya di dalam dokumen.



6. Simbol asrip permanen

Menunjukkan tempat penyimpanan dokumen yang tidak akan di proses lagi.



7. Simbol garis alur

Menunjukkan alur di proses.



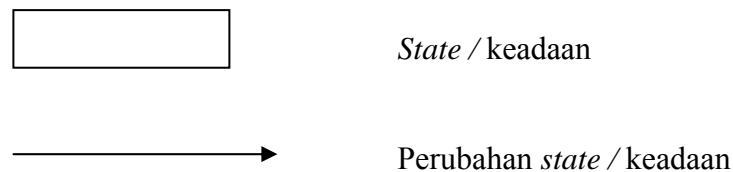
8. Simbol terminal

Menunjukkan sumber atau tujuan dari dokumen atau laporan.



2.2.8.5 State Transition Diagram (STD)

State Transition Diagram (STD) merupakan suatu *tools* permodelan yang menggambarkan sifat ketergantungan pada waktu dari suatu sistem. Adapun simbol yang digunakan :



Untuk melengkapi STD masih diperlukan dua hal lagi yaitu :

1. *Condition* (kondisi), adalah sebuah sinyal yang menyebabkan perubahan terhadap *state* dari *state* ke *state* berikutnya.
2. *Action* (aksi), adalah yang dilakukan sistem bila terjadi perubahan *state* atau merupakan reaksi terhadap kondisi.

2.3 Konsep Interaksi Manusia dan Komputer

2.3.1 Definisi IMK dan Tujuan Perancangan *User-Interface*

Menurut Hewett, et, al. (1996, p.5) dalam bukunya yang berjudul *ACM SIGCHI, Curricula for Human-Computer Interaction*, dinyatakan bahwa Interaksi Manusia dan Komputer (IMK) atau *Human-Computer Interaction* (HCI) adalah disiplin ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia, serta *studi* fenomena-fenomena besar yang berhubungan dengannya.

Sedangkan ACM SIGCHI sendiri merupakan kependekan dari *The Association for Computing Machinery-Special Interest Group on Computer Human* (<http://www.sigchi.org>)

Menurut Shneiderman (1998, pp. 14-16), tujuan perancangan *user-interface* adalah untuk menentukan komunitas user dan menandakan tugas – tugasnya untuk memenuhi tujuan manusia. Perlu diingat bahwa desain yang baik bagi komunitas yang satu, bisa saja tidak sesuai bagi komunitas lainnya, dan desain yang efisien bagi sekumpulan tugas, bisa saja tidak efisien bagi kumpulan tugas lainnya.

2.3.2 Prinsip – prinsip Perancangan *User-Interface*

Menurut Shneiderman dalam bukunya *Designing the User Interface, Strategies for Effective Human-Computer Interaction* (1998, pp 67-79), ada 3 prinsip dalam merancang *user-interface*.

Prinsip 1 : Kenali perbedaan berdasarkan jenis – jenis pemakai:

a. *Novice (first-time users)*

Konsep tugas dan *interface* dangkal.

Perancangan:

1. Batasi jumlah pilihan,
2. Umpan balik yang informatif,
3. *Manual* dan petunjuk *online* yang efektif.

b. *Knowledgeable intermittent users*

Konsep tugas stabil.

Konsep *interface* luas namun sulit mengingat secara sintaktik.

Perancangan:

1. Struktur menu yang rapi,
2. Konsisten,
3. Kejelasan *interface* yang jelas,
4. Perlindungan dari bahaya karena eksplorasi fitur.

c. *Expert frequent users*

Terbiasa dengan konsep tugas dan *interface*, ingin pekerjaan cepat selesai.

Perancangan

1. Makro,
2. *Shortcuts*,
3. Singkatan, dsb.

Prinsip 2 : Gunakan Delapan Aturan Emas Perancangan *User-Interface*

1) Berusaha untuk konsisten.

Website yang dirancang harus konsisten, baik dalam hal tampilan layar, susunan menu, teks, dan warna.

2) Memungkinkan *frequent users* menggunakan *shortcuts*.

Website yang dirancang sebaiknya memiliki fasilitas *shortcut* bagi *user* agar dapat menjelajah *website*.

3) Memberi umpan balik yang informatif.

Sebuah *website* harus dapat menyediakan navigasi atau informasi mengenai tujuan dan suatu *link*, untuk mengurangi kemungkinan terjadinya kesalahan dan memberikan kepuasan bagi *user*.

4) Merancang dialog yang memberikan penutupan (keadaan akhir).

Aksi – aksi yang ada seharusnya diorganisasikan agar memiliki suatu awal, pertengahan, dan tahap akhir. Dengan adanya umpan balik yang informatif pada tahap akhir dari suatu halaman akan memberitahukan *user* sehingga dapat mengetahui kapan *user* dapat berpindah ke halaman berikutnya.

5) Memberikan pencegahan kesalahan dan penanganan kesalahan yang sederhana.

Jika terjadi kesalahan, maka *website* mampu memberikan petunjuk sederhana dan praktis dalam menanganinya.

6) Memungkinkan pembalikan aksi yang mudah.

Website harus dapat menyediakan fasilitas bagi *user* untuk kembali ke menu semula dengan mudah.

7) Mendukung pusat kendali internal

Seorang *user* yang terhubung dengan jaringan selalu mempunyai tujuan. Biarkan *user* memutuskan apa yang diperlukan dan kemudian menyediakan informasi yang dibutuhkan.

8) Mengurangi beban ingatan jangka pendek

Website harus memudahkan *user* dalam mengingat hal-hal penting, misalnya dengan kombinasi kode, maka kode tersebut diusahakan mudah diingat dengan kemampuan berpikir manusia, hindari kode yang terlalu panjang.

Prinsip 3 : Cegah Kesalahan

- Membetulkan pasangan yang bersesuaian,
- Melengkapi urutan aksi,
- Membetulkan perintah.

2.4 Personal Home Page (PHP)

Menurut Nugroho (2004, p 29), PHP adalah bahasa standar yang digunakan dalam dunia *website*. PHP adalah bahasa program yang berbentuk *script* yang diletakkan di dalam *server web*. Jika dilihat dari sejarah, mulanya PHP diciptakan oleh ide Rasmus Lerdof yang membuat *script perl*. *Script* tersebut sebenarnya dimaksudkan untuk digunakan sebagai program untuk dirinya sendiri. Akan tetapi kemudian dikembangkan lagi sehingga menjadi bahasa yang disebut “Personal Home Page”. Inilah awal mula munculnya PHP sampai saat ini.

2.5 MySQL

Menurut Nugroho (2004, p 29), MySQL adalah sebuah program pembuat database yang bersifat *open source*, yang artinya siapa saja boleh menggunakannya dan tidak akan dicekal. MySQL juga merupakan program pengakses database yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multi users* (banyak pengguna). Kelebihan lainnya dari MySQL adalah bahasa *query* standar yang dimiliki SQL (*Structure Query Language*)

2.6 Macromedia Dreamweaver MX

Macromedia Dreamweaver MX adalah sebuah HTML editor profesional untuk mendisain secara visual dan mengelola situs *web* maupun halaman *web*. Dreamweaver membuatnya jadi lebih mudah dengan menyediakan *tool* yang sangat berguna dalam meningkatkan kemampuan dan pengalaman membuat *web*.

Dreamweaver juga mengikutsertakan banyak *tool* untuk kode-kode dalam hal *web* beserta fasilitas-fasilitasnya antara lain : Referensi HTML, CSS, *Java Script Debugger*, dan *Editor Code* yang mengizinkan pengeditan kode *JavaScript XML*, dan dokumen teks lain secara langsung dalam Dreamweaver.

2.7 Teori –teori persediaan dan penjualan

2.7.1 Persediaan

Menurut Mulyadi (2001, p 553), sistem persediaan bertujuan untuk mencatat mutasi tiap jenis persediaan yang disimpan di gudang. Sistem ini berkaitan erat dengan sistem penjualan, sistem retur penjualan, sistem pembelian, sistem retur pembelian, dan sistem akuntansi biaya produksi. Dalam perusahaan manufaktur, persediaan terdiri dari persediaan produk jadi, persediaan suku cadang. Dalam perusahaan dagang, persediaan hanya terdiri dari satu golongan yaitu persediaan barang dagangan yang merupakan barang yang akan dibeli dengan tujuan untuk dijual kembali.

Menurut Mulyadi (2001, p 556), ada 2 macam metode pencatatan persediaan, yaitu metode mutasi persediaan dan metode persediaan fisik. Dalam

metode mutasi persediaan, setiap mutasi persediaan dari pembelian saja yang dicatat, sedangkan mutasi berkurangnya persediaan karena pemakaian tidak dicatat dalam kartu persediaan.

Menurut Mulyadi (2001, p 578), fungsi yang terkait dalam sistem perhitungan fisik persediaan, yaitu :

1. Panitia perhitungan fisik persediaan berfungsi untuk melaksanakan perhitungan fisik persediaan dan menyerahkan hasil perhitungan tersebut kepada bagian kartu persediaan untuk digunakan sebagai dasar penyesuaian terhadap catatan persediaan dalam kartu persediaan.
2. Fungsi akutansi bertanggung jawab untuk :
 - a. Mencantumkan harga pokok sistem persediaan yang dihitung dalam daftar hasil perhitungan fisik.
 - b. Mengalikan kuantitas dan harga pokok per satuan yang tercantum dalam hasil perhitungan fisik.
 - c. Mencantumkan harga pokok total dalam daftar hasil perhitungan fisik.
 - d. Melakukan penyesuaian terhadap kartu persediaan berdasarkan data hasil perhitungan fisik persediaan.
 - e. Membuat bukti memorial untuk mencatat penyesuaian data persediaan dalam jurnal umum berdasarkan hasil perhitungan fisik persediaan.
3. Fungsi gudang bertanggung jawab untuk melakukan penyesuaian data kuantitas persediaan yang dicatat dalam kartu gudang berdasarkan hasil perhitungan fisik persediaan.

Menurut Mulyadi (2001, p 559), sistem dan prosedur yang bersangkutan dengan sistem persediaan adalah :

1. Prosedur pencatatan produk jadi

Prosedur ini merupakan salah satu prosedur dalam sistem akutansi biaya produksi. Dalam prosedur ini dicatat harga pokok produksi jadi yang didebitkan ke dalam rekening Persediaan Produk Jadi dan dikreditkan ke dalam rekening

Barang Dalam Proses. Catatan akutansi yang digunakan dalam prosedur pencatatan produk jadi adalah kartu gudang, kartu persediaan, dan jurnal umum.

2. Prosedur pencatatan harga pokok produk jadi yang dijual

Prosedur ini merupakan salah satu prosedur dalam sistem penjualan di samping prosedur lainnya seperti prosedur order penjualan, prosedur persetujuan kredit, prosedur pengiriman barang, prosedur penagihan, prosedur pencatatan piutang. Catatan akutansi yang digunakan dalam prosedur pencatatan harga pokok produk jadi yang dijual adalah kartu gudang, kartu persediaan, jurnal umum.

3. Prosedur pencatatan harga pokok produk jadi yang diterima kembali dari pembeli

Prosedur ini merupakan salah satu prosedur yang membentuk sistem retur barang. Jika produk jadi yang telah dijual dikembalikan oleh pembeli, maka transaksi retur penjualan ini akan mempengaruhi persediaan produk jadi. Catatan akutansi yang digunakan dalam prosedur pencatatan produk jadi adalah kartu gudang, kartu persediaan, dan jurnal umum atau jurnal retur persediaan, jika perusahaan menggunakan jurnal khusus.

4. Prosedur pencatatan tambahan dan penyesuaian kembali harga pokok persediaan produk dalam proses

Pencatatan persediaan produk dalam proses biasanya dilakukan oleh perusahaan pada akhir periode, pada saat dibuat laporan keuangan bulanan dan laporan keuangan tahunan. Pencatatan persediaan produk dalam jurnal umum.

5. Prosedur pencatatan harga pokok persediaan yang dibeli

Prosedur ini merupakan prosedur yang membentuk sistem pembelian. Dalam prosedur ini dicatat harga pokok persediaan yang dibeli.

6. Prosedur pencatatan harga pokok persediaan yang dikembalikan kepada pemasok

Prosedur ini merupakan salah satu prosedur yang membentuk sistem retur pembelian. Jika persediaan yang telah dibeli dikembalikan kepada pemasok, maka transaksi retur pembelian ini akan mempengaruhi persediaan yang bersangkutan.

7. Prosedur permintaan dan pengeluaran barang gudang

Prosedur ini merupakan salah satu prosedur yang membentuk sistem akuntansi biaya produksi. Dalam prosedur ini dicatat harga pokok persediaan bahan baku, bahan penolong, bahan habis pakai pabrik, dan suku cadang yang dipakai dalam kegiatan produksi dan kegiatan non produksi.

8. Prosedur pencatatan tambahan harga pokok persediaan karena pengembalian barang gudang

Prosedur ini melakukan transaksi pengembalian barang gudang mengurangi biaya dan menambah persediaan barang di gudang. Jurnal yang dibuat untuk mencatat transaksi tersebut dalam jurnal umum.

9. Sistem perhitungan fisik persediaan

Sistem perhitungan fisik persediaan umumnya digunakan oleh perusahaan untuk menghitung secara fisik persediaan yang disimpan di gudang, yang hasilnya digunakan untuk meminta pertanggungjawaban. Bagian gudang mengenai pelaksanaan fungsi penyimpanan, dan pertanggungjawaban bagian kartu persediaan mengenai keandalan catatan persediaan yang diselenggarakan, serta untuk melakukan penyesuaian terhadap catatan persediaan di bagian Kartu Persediaan.

2.7.2 Penjualan

Menurut Mulyadi (2001, p 202), kegiatan penjualan terdiri dari penjualan barang dan jasa, baik secara kredit maupun secara tunai. Dalam transaksi

penjualan kredit jika order dari pelanggan telah dipenuhi dengan permintaan barang atau penyerahan jasa, untuk jangka waktu tertentu perusahaan memiliki piutang kepada pelanggannya. Dalam sistem penjualan secara tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli.

Menurut Mulyadi (2001, p 462), fungsi yang terkait dengan penerimaan kas dari penjualan secara tunai adalah :

1. Fungsi penjualan

Fungsi ini bertanggung jawab untuk menerima order dari pembeli, mengisi faktur penjualan tunai, dan menyerahkan faktur tersebut kepada pembeli untuk kepentingan pembayaran ke fungsi kas.

2. Fungsi kas

Fungsi ini bertanggung jawab sebagai penerimaan kas dari pembeli.

3. Fungsi gudang

Fungsi ini bertanggung jawab untuk menyiapkan barang yang dipesan oleh pembeli, serta menyerahkan barang tersebut ke fungsi pengiriman.

4. Fungsi pengiriman

Fungsi ini bertanggung jawab untuk membungkus barang dan menyerahkan barang yang telah dibayar harganya kepada pembeli.

5. Fungsi akuntansi

Fungsi ini bertanggung jawab sebagai pencatat transaksi penjualan dan penerimaan kas serta pembuat laporan penjualan.

Menurut Mulyadi (2001, p 469), jaringan prosedur yang membentuk sistem penerimaan kas dari penjualan tunai :

1. Prosedur order penjualan

Dalam prosedur ini fungsi penjualan menerima order dari pembeli dan membuat faktur penjualan tunai untuk memungkinkan pembeli melakukan pembayaran harga barang ke fungsi kas dan untuk memungkinkan fungsi gudang fungsi pengiriman menyiapkan barang yang akan diserahkan kepada pembeli.

2. Prosedur penerimaan kas

Dalam prosedur ini fungsi kas menerima pembayaran harga barang dari pembeli dan memberikan tanda pembayaran kepada pembeli untuk memungkinkan pembeli tersebut melakukan pengambilan barang yang dibelinya dari fungsi pengiriman.

3. Prosedur penyerahan barang

Dalam prosedur ini fungsi pengiriman menyerahkan barang kepada pembeli.

4. Prosedur pencatatan penjualan tunai

Dalam prosedur ini fungsi akuntansi melakukan pencatatan transaksi penjualan tunai dalam jurnal penjualan dan jurnal penerimaan kas.

5. Prosedur penyetoran kas ke bank

Dalam prosedur ini fungsi kas menyetor kas yang diterima dari penjualan tunai ke bank dalam jumlah penuh.

6. Prosedur pencatatan penerimaan kas

Dalam prosedur ini, fungsi akuntansi mencatat penerimaan kas ke dalam jurnal penerimaan kas berdasarkan bukti setor bank yang diterima dari bank melalui fungsi kas.

7. Prosedur pencatatan harga pokok penjualan

Dalam prosedur ini, fungsi akuntansi membuat rekapitulasi harga pokok penjualan berdasarkan data yang dicatat dalam kartu persediaan.