BAB 2

LANDASAN TEORI

2.1 Pendekatan Basis Data

Teori-teori dasar yang berkaitan dengan basis data dan digunakan untuk menunjang proses pembuatan skripsi ini ialah teori mengenai Data, Basis Data (*Database*), Sistem Basis Data (*Database System*), Sistem Manajemen Basis Data (*Database Management System*- DBMS), *Database Language*, Siklus Hidup Aplikasi Basis Data (*Database Application Life Cycle*), Teknik Pengumpulan Fakta (*Fact Finding Techniques*), Pemodelan Hubungan Entitas (*Entity Relationship Modelling*), Metode Perancangan Basis Data (*Database Design Methodology*), dan Normalisasi (*Normalization*).

2.1.1 Data

Menurut Whitten, J. L., Bentley, L. D., dan Dittman, K. A. (2004), data adalah fakta mentah tentang orang, tempat, kejadian, dan hal-hal penting pada perusahaan.

Mengacu pada pendapat Connolly dan Begg (2010), data adalah komponen terpenting yang harus dimiliki dalam DBMS, dimana data tersebut dapat diperoleh dari *end-users*. Semua data operasional perlu dimasukkan ke dalam DBMS karena data tersebut nantinya akan diolah untuk memenuhi kebutuhan informasi organisasi.

Dengan kata lain, data merupakan fakta mentah, yang belum diolah dan berasal dari *end-users* serta operasional perusahaan, dimana fakta tersebut nantinya dapat diolah untuk menghasilkan informasi yang berguna bagi perusahaan.

2.1.2 Basis Data (*Database*)

Menurut Elmasri dan Navathe (2004) basis data adalah kumpulan dari data yang saling berhubungan, dimana data merupakan fakta yang dapat dicatat dan memiliki arti yang implisit. Misalnya saja, kumpulan dari nama, nomor telepon, dan alamat dari orang-orang, disimpan dalam buku alamat atau dalam *hard drive*, yang menggunakan bantuan komputer.

Mengacu pada pendapat Connolly dan Begg (2010) basis data dapat didefiniskan sebagai sekumpulan data logis yang saling berhubungan dan dideskripsikan, dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

Berbeda dengan *system file* menyimpan data secara terpisah, basis data menyimpan seluruh data dalam sebuah *repository* besar secara terintegrasi. Dengan demikian, basis data tidak hanya diperuntukkan untuk suatu departemen, melainkan menjadi sumber daya yang dapat dimanfaatkan oleh seluruh departemen dalam sebuah organisasi.

2.1.3 Sistem Basis Data (*Database System*)

Menurut O'Brien (2003) sistem basis data adalah serangkaian program komputer yang berfungsi untuk mengendalikan pembuatan, pemeliharaan, serta pemanfaatan basis data dalam suatu organisasi.

Menurut Connolly dan Begg (2010) sistem basis data merupakan komponen fundamental dari suatu organisasi besar dengan sistem informasi yang luas. Sistem basis data adalah sekumpulan program atau perangkat lunak yang berhubungan dengan basis data.

Dengan bantuan sistem basis data, pengguna dapat mendefinisikan, membuat, memelihara, dan mengendalikan akses ke dalam basis data. Tiap organisasi yang besar dengan jumlah data yang banyak dapat memanfaatkan sistem basis data untuk menunjang sistem dan proses bisnis yang berjalan.

2.1.4 Sistem Manajemen Basis Data (*Database Management System - DBMS*)

Berikut ini adalah pengertian DBMS, komponen *DBMS environment*, fungsi DBMS, serta keuntungan dan kerugian penggunaan DBMS.

2.1.4.1 Pengertian DBMS

Berpedomankan pada pendapat Connolly dan Begg (2010), sistem manajemen basis data dapat didefinisikan sebagai sebuah sistem *software* yang memungkinkan pengguna untuk

mendefinisikan, membuat, memelihara, dan mengontrol akses terhadap basis data.

Berikut ini merupakan fasilitas-fasilitas yang dimiliki oleh DBMS:

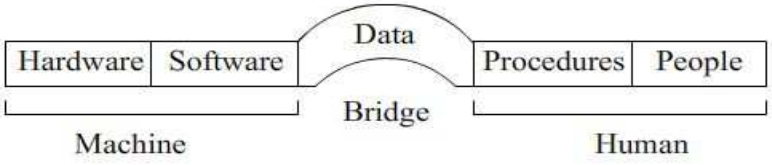
- Dengan menggunakan DBMS, pengguna dapat mendefinisikan basis data melalui *Data Definition Language* (DDL). Berdasarkan pendapat Connolly dan Begg (2010) DDL mengijinkan spesifikasi tipe data, struktur, dan batasan (*constraint*) untuk data yang disimpan di dalam basis data.
- Mengacu pada pendapat Connolly dan Begg (2010) DBMS memungkinkan pengguna untuk melakukan operasi *insert*, *update*, *delete*, dan mengambil (*retrieve*) data dari basis data.
- Di samping menyediakan DDL dan DML, menurut Connolly dan Begg (2010), DBMS juga menyedikan akses kontrol ke dalam basis data, antara lain:
    a. Sistem keamanan (*security system*), yang mencegah basis data diakses oleh pihak yang tidak berkepentingan dan tidak memiliki hak akses,
    b. Sistem integritas (*integrity system*), yang menjaga konsistensi data, sehingga tidak ada lagi redudansi data,
    c. Sistem kontrol konkurensi (*concurrency control system*), yang memungkinkan banyak pengguna dapat mengakses basis data dalam waktu yang bersamaan,

d. *Recovery control system*, yang menyimpan basis data secara berkala pada media penyimpanan yang konsisten,

e. *User-accessible catalog*, yang menyimpan deskripsi data dalam basis data.

2.1.4.2 Komponen *DBMS Environment*

Mengacu pada pendapat Connolly dan Begg (2010) terdapat lima komponen utama dalam lingkungan DBMS, yaitu (Gambar 2.1):



Gambar 2.1 *DBMS Environment*

Sumber: Connolly dan Begg, 2010

- Perangkat keras (*hardware*)

  DBMS dan aplikasinya memerlukan perangkat keras agar dapat beroperasi. Perangkat keras meliputi *personal computer* (PC) sampai *mainframe* atau jaringan dari beberapa komputer.

- Perangkat lunak (*software*)

  Komponen perangkat lunak terdiri dari perangkat lunak DBMS itu sendiri dan program aplikasi, bersama dengan sistem operasi, termasuk perangkat lunak jaringan apabila DBMS digunakan pada jaringan yang tersusun dari beberapa komputer.

- Data

  Data merupakan komponen yang paling penting dalam lingkungan DBMS, terutama jika dilihat dari sudut pandang *end-users*. Data berfungsi untuk menjembatani komponen mesin dan komponen manusia.

- Prosedur (*procedures*)

  Prosedur adalah instruksi dan aturan yang dimasukkan ke dalam program agar dapat mengatur rancangan dan penggunaan basis data. Contoh prosedur agar dapat menggunakan DBMS ialah instruksi *log on* ke dalam DBMS, menggunakan fasilitas DBMS atau aplikasi tertentu, memulai dan mengakhiri DBMS, membuat *backup* yang berisi salinan basis data, dan mengubah struktur tabel.

- Manusia (*people*)

  Komponen akhir yang digunakan pada DBMS adalah semua orang yang terlibat dalam sistem, seperti *Data Administrator* (DA), *Database Administrator* (DBA), *Database Designers*, *Application Developers*, dan *End-Users*.

2.1.4.3 Fungsi DBMS

Menurut Connolly dan Begg (2010) terdapat sepuluh fungsi dasar yang harus dimiliki oleh DBMS, yaitu:

1. *Data storage*, *retrieval*, dan *update*

   DBMS harus menyediakan kemampuan untuk menyimpan, mengambil, dan meng-*update* data pada basis data.

2. *User-accessible catalog*

   DBMS harus menyediakan katalog yang menyimpan deskripsi *data items* dapat diakses oleh pengguna.

3. *Transaction support*

   DBMS harus menyediakan mekanisme untuk memastikan tiap *update* yang berhubungan dengan transaksi telah dilakukan atau sama sekali belum pernah dilakukan.

4. *Concurrency control services*

   DBMS harus menyediakan mekanisme untuk memastikan bahwa basis data telah di-*update* dengan benar ketika banyak pengguna melakukan *update* secara bersamaan.

5. *Recovery services*

   DBMS harus menyediakan mekanisme untuk melakukan perbaikan (*recovery*) ketika terjadi kerusakan pada basis data.

6. *Authorization services*

   DBMS harus menyediakan mekanisme untuk memastikan basis data hanya dapat diakses oleh pihak yang memiliki hak akses.

7. *Support for data communication*

   DBMS harus dapat diintegrasikan dengan perangkat lunak komunikasi (*communication software*).

8. *Integrity services*

   DBMS harus menyediakan suatu sarana untuk memastikan tiap data dalam basis data dan perubahan pada data mengikuti aturan-aturan tertentu.

9. *Services to promote data independence*

   DBMS harus menyediakan fasilitas untuk mendukung agar program dapat berdiri sendiri, tanpa bergantung pada struktur basis data.

10. *Utility services*

    DBMS harus menyediakan sekumpulan fitur pendukung (*utility services*) yang mampu membantu DBA dalam mengelola basis data secara efektif.

2.1.4.4 Keuntungan dan Kerugian Penggunaan DBMS

Berdasarkan pendapat Connolly dan Begg (2010) dapat disimpulkan keuntungan dan kerugian penggunaan DBMS adalah sebagai berikut (Tabel 2.1):

Tabel 2.1 Keuntungan dan Kerugian Penggunaan DBMS

Keuntungan	Kerugian
• Kontrol redudansi data • Konsistensi data • Informasi yang lebih banyak dari sejumlah data yang sama	• Kompleksitas • Ukuran (*size*) • Biaya DBMS • Biaya *hardware* tambahan

Keuntungan	Kerugian
• *Sharing data* • Integritas data meningkat • Kemanan meningkat • *Enforcement of standard* • *Economy of scale* • *Balance of conflicting requirements* • *Data accessibility and responsiveness* meningkat • Produktifitas meningkat • *Maintenance* yang meningkat akibat adanya *data independence* • Konkurensi meningkat Pelayanan *backup* dan *recovery* meningkat	• Biaya konversi • Kinerja (*performance*) • Dampak kegagalan lebih tinggi

2.1.5 *Database Language*

Menurut Connolly dan Begg (2010) *data sublanguage* terdiri dari dua bagian, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). DDL digunakan untuk membuat spesifikasi skema basis

data. Sedangkan DML digunakan untuk membaca dan meng-*update* basis data.

2.1.5.1 *Data Definition Language* (DDL)

Menurut Connolly dan Begg (2010) DDL merupakan suatu bahasa yang memungkinkan DBA atau pengguna untuk mendeskripsikan dan memberikan nama kepada entitas, atribut, dan relasi yang dibutuhkan pada aplikasi, serta *integrity* dan *security constraints*.

Hasil dari DDL statement berupa sekumpulan tabel yang disimpan dalam *file* terstruktur yang disebut dengan sistem katalog. Kumpulan sistem katalog inilah yang membentuk metadata, yang merupakan data yang mendeskripsikan objek dalam basis data dan membuat objek tersebut dapat lebih mudah diakses dan dimanipulasi.

2.1.5.2 *Data Manipulation Language* (DML)

Berdasarkan pendapat Connolly dan Begg (2010) DML adalah bahasa yang menyediakan sekumpulan operasi untuk mendukung manipulasi dasar pada data yang disimpan di dalam basis data. Berikut ini adalah beberapa operasi pada DML:

a. Penambahan (*insertion*) data baru ke dalam basis data,
b. Pemodifikasian (*modification*) data yang disimpan di dalam basis data,

c. Pengambilan (*retrieval*) data yang disimpan di dalam basis data,

d. Penghapusan (*deletion*) data dari dalam basis data

2.1.5.3 *Fourth-Generation Languages* (4GLs)

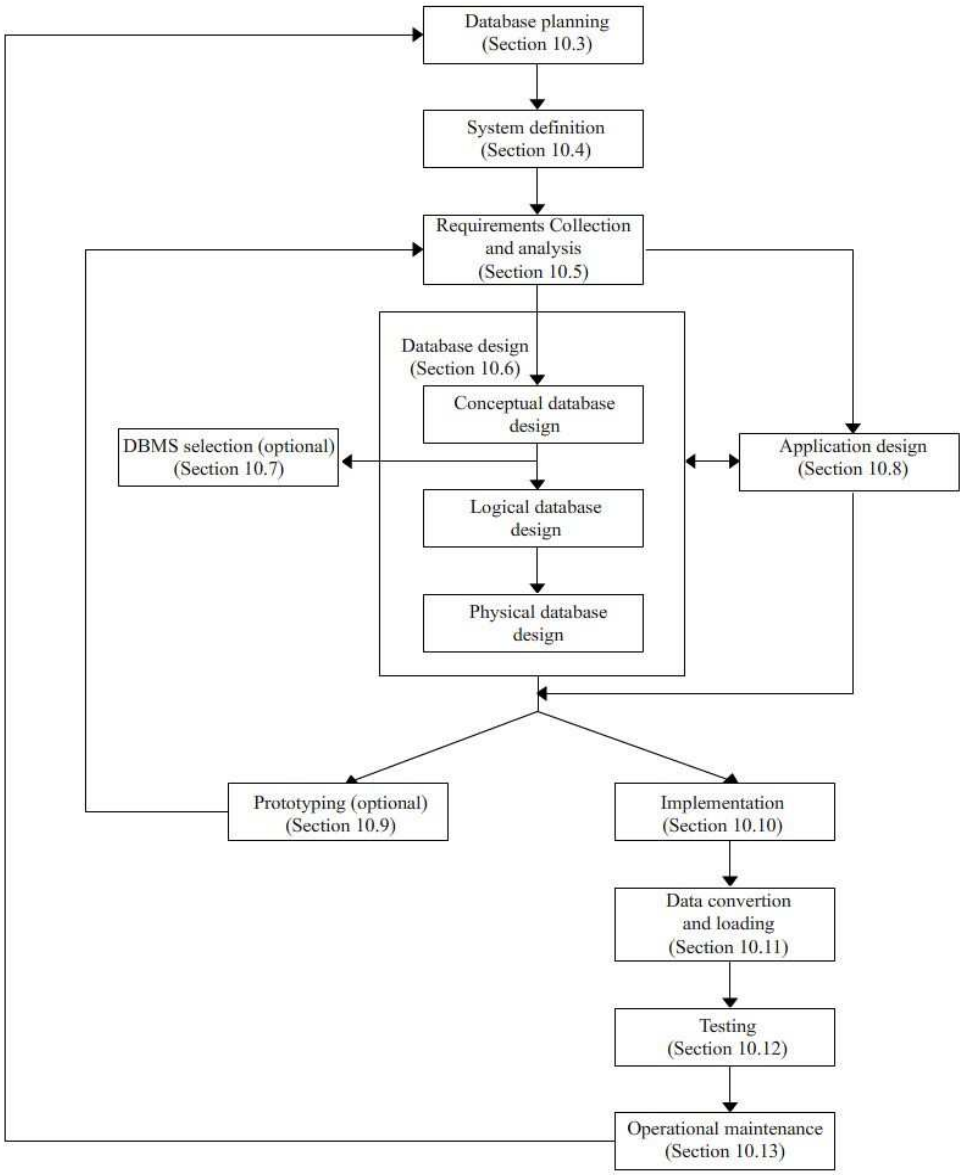
Menurut Connolly dan Begg (2010) 4GLs biasa disebut *shorthand programming language*. Sebuah operasi yang membutuhkan ratusan baris pada 3GL, pada umumnya membutuhkan baris yang jauh lebih sedikit jumlahnya pada 4GLs.

4GLs mencakup beberapa hal, yaitu:

- *presentation language*, seperti *query language* dan *report generator*,
- *speciality language*, seperti *spreadsheets* dan bahasa basis data,
- *application generator*, yang mendefinisikan, *insert*, *update*, dan *retrieve* data dari basis data untuk membangun aplikasi,
- *very high-level language*, yang digunakan untuk menghasilkan *application code*.

2.1.6 Siklus Hidup Aplikasi Basis Data (*Database Application Life Cycle*)

Menurut Connolly dan Begg (2010), siklus hidup sistem basis data adalah pendekatan terstruktur yang terdiri dari tahap-tahap dalam pengembangan sistem basis data. Tahap-tahap tersebut terdiri dari (Gambar 2.2):



Gambar 2.2 Siklus Hidup Aplikasi Basis Data

Sumber: Connolly dan Begg, 2010

2.1.6.1 Perencanaan Basis Data (*Database Planning*)

Menurut Connolly dan Begg (2010) *database planning* adalah kegiatan perencanaan dan pengaturan agar seluruh tahapan dalam siklus hidup sistem basis data dapat direalisasikan seefektif

dan seefisien mungkin. Langkah-langkah yang perlu dilakukan dalam perencanaan basis data terdiri dari:

- Menentukan *mission statement*

  Mission statement membantu menjelaskan tujuan dari sistem basis data dan menyediakan alur yang lebih jelas agar dapat menciptakan sistem basis data dengan efektif dan efisien.

- Menentukan *mission objectives*

  Mission objectives perlu diidentifikasi setelah *mission statement* ditentukan. Setiap *mission objective* harus dapat mengidentifikasi tugas tertentu yang dapat dilakukan oleh sistem basis data.

2.1.6.2 Pendefinisian Sistem (*System Definition*)

Menurut Connolly dan Begg (2010) *system definition* menjelaskan batasan dan ruang lingkup dari sistem basis data dan garis besar dari sudut pandang pengguna (*user views*). Sebelum merancang sistem basis data, sangat penting untuk mengidentifikasikan terlebih dahulu batasan dari sistem yang sedang dibangun, serta bagaimana sistem berinteraksi dengan bagian-bagian lain dari sistem informasi organisasi.

2.1.6.3 Pengumpulan dan Penganalisisan Kebutuhan (*Requirement Collection and Analysis*)

Menurut Connolly dan Begg (2010) *requirements collection and analysis* adalah proses pengumpulan dan penganalisisan informasi tentang bagian dari organisasi yang harus didukung oleh sistem basis data dan dengan menggunakan sistem informasi tersebut mengidentifikasi data yang diperlukan untuk membangun sistem yang baru. Tahap ini dapat dilakukan dengan menggunakan banyak teknik, yang disebut dengan *fact finding techniques*.

2.1.6.4 Perancangan Basis Data (*Database Design*)

Mengacu pada pendapat Connolly dan Begg (2010) *database design* adalah proses pembuatan rancangan sistem basis data yang mendukung *mission statement* dan *mission objective* perusahaan.

Perancangan basis data terdiri dari tiga tahapan utama, yaitu perancangan basis data konseptual (*conceptual database design*), logikal (*logical database design*), dan fisikal (*physical database design*).

Terdapat beberapa pendekatan dalam perancangan basis data antara lain:

- Pendekatan *bottom-up*

  Menurut Connolly dan Beg (2010) pendekatan *bottom-up* dimulai dari level paling dasar yaitu atribut (properti dari entitas dan relasi), kemudian dilakukanlah analisis terhadap hubungan antaratribut, setelah itu dilakukan pengelompokkan berdasarkan relasi yang merepresentasikan entitas dan relasi antarentitas. Contoh pendekatan ini ialah dalam melakukan normalisasi.

- Pendekatan *top-down*

  Menurut Connolly dan Begg (2010) pendekatan *top-down* dimulai dengan pengembangan model data tingkat tinggi, yang dimulai dengan mengidentifikasi entitas, relasi, dan atribut yang terkait. Pendekatan ini diilustrasikan dalam konsep *Entity-Relationship (ER) model*.

- Pendekatan *inside-out*

  Mengacu pada pendapat Connolly dan Begg (2010) *inside out* adalah pendekatan yang berhubungan dengan *bottom-up*, namun perbedaannya adalah pada *inside-out* tahap pertama dimulai dengan pengidentifikasian sekumpulan entitas utama (*major entities*), kemudian diturunkan menjadi entitas, relasi, dan atribut yang berhubungan dengan entitas utama.

- Pendekatan *mixed strategy*

  Berdasarkan pada pendapat Connolly dan Begg (2010) *mixed strategy* adalah pendekatan gabungan antara *bottom-up* dan *top-down* sebelum akhirnya menggabungkan semua bagian.

2.1.6.5 Pemilihan DBMS (*DBMS Selection*) - *optional*

Menurut Connolly dan Begg (2010) *DBMS selection* adalah proses pemilihan DBMS yang tepat untuk mendukung sistem basis data. Pendekatan sederhana untuk memilih DBMS ialah dengan memeriksa apakah fitur-fitur yang disediakan oleh DBMS mampu memenuhi kebutuhan sistem basis data yang akan dirancang.

2.1.6.6 Perancangan Aplikasi (*Application Design*)

Mengacu pada pendapat Connolly dan Begg (2010) application design adalah perancangan antarmuka (*user interface*) dan program aplikasi yang menggunakan dan memproses basis data. Rancangan antarmuka dan program aplikasi harus disesuaikan dengan kebutuhan pengguna.

2.1.6.7 *Prototyping* (*optional*)

Menurut Connolly dan Begg (2010) *prototyping* adalah pembuatan kerangka kerja (*working model*) untuk sistem basis

data. Tahap ini dilakukan agar pengguna dapat mengidentifikasi apakah fitur-fitur sistem berjalan dengan baik, dan jika mungkin pengguna dapat memberi fitur tambahan pada sistem basis data.

2.1.6.8 Implementasi (*Implementation*)

Mengacu pada pendapat Connolly dan Begg (2010) *implementation* adalah realisasi fisik dari basis data dan rancangan aplikasi. Implementasi basis data dapat dicapai dengan menggunakan DDL dari DBMS. Program aplikasi dapat diimplementasikan degan menggunakan 3GL atau 4GL.

2.1.6.9 *Data Conversion and Loading*

Menurut Connolly dan Begg (2010) *data conversion* and *loading* adalah pemindahan data ke dalam basis data yang baru dan pengkonversian aplikasi agar dapat berjalan pada basis data yang baru. Tahap ini dibutuhkan jika sistem basis data yang baru akan menggantikan sistem yang lama.

2.1.6.10 Pengujian (*Testing*)

Berdasarkan pada pendapat Connolly dan Begg (2010) *testing* adalah proses pengeksekusian sistem basis data dengan tujuan agar dapat menemukan kesalahan (*error*).

2.1.6.11 Perawatan Operasional (*Operational Maintenance*)

Mengacu pada pendapat Connolly dan Begg (2010) *operational maintenance* adalah proses pengawasan (*monitoring*) dan pemeliharaan (*maintaining*) sistem basis data setelah instalasi.

2.1.7 Teknik Pengumpulan Fakta (*Fact Finding Techniques*)

Teknik pengumpulan fakta adalah teknik yang digunakan untuk memperoleh fakta dan data yang dibutuhkan dalam sistem basis data. Menurut Connolly dan Begg (2010) terdapat lima teknik pengumpulan fakta yang cukup sering digunakan, yaitu:

- Pemeriksaan dokumentasi (*examining documentation)*

  Teknik ini dapat dilakukan dengan memeriksa dokumen, *form*, laporan, dan *files* yang berhubungan dengan sistem lama. Dengan menggunakan teknik ini akan diperoleh pemahaman akan sistem dengan cepat.

- Wawancara (*interviewing*)

  Wawancara adalah teknik yang paling sering digunakan dan pada umumnya paling berguna dalam menemukan fakta dan data yang diperlukan. Wawancara dapat dilakukan ke pengguna sistem. Teknik ini terdiri dari wawancara terstruktur, dimana pertanyaan wawancara sudah disiapkan sebelumnya, dan wawancara tidak terstruktur, dimana hanya ada sedikit atau bahkan tidak ada daftar pertanyaan yang spesifik.

  Berdasarkan respon dari pihak yang diwawancara (*interviewee*), wawancara dibedakan menjadi *open-ended questions* dan *close-ended*

questions. Pada wawancara dengan *open-ended questions* pewawancara (*interviewer*) tidak memberikan pilihan jawaban yang spesifik. Sedangkan pada wawancara dengan *close-ended questions* pewawancara telah menyediakan pilihan jawaban atas pertanyaan yang diajukan.

- Observasi operasional perusahaan (*observing the enterprise in operation*)

  Observasi operasional perusahaan adalah teknik yang paling efektif dalam pengumpulan fakta karena dengan melakukan observasi, sistem akan lebih mudah dimengerti. Observasi operasional perusahaan dapat dilakukan dengan mengikuti kegiatan semua divisi perusahaan, contohnya dengan melakukan kerja praktek pada perusahaan yang ingin diamati.

- Penelitian (*research*)

  Dengan melakukan penelitian sistem dan masalah yang terjadi pada sistem menjadi mudah untuk diamati. Penelitian dapat dilakukan tanpa perlu datang langsung ke perusahaan terkait karena teknik ini mencari informasi berdasarkan data yang sudah pernah terjadi. Penelitian memperoleh data dengan bersumberkan pada jurnal, buku referensi, dan internet.

- Kuesioner (*questionnaires*)

  Kuesioner adalah dokumen yang dibuat dengan tujuan khusus untuk memungkinkan fakta dapat dikumpulkan dari sekelompok orang dalam jumlah besar. Berdasarkan jenis pertanyaan, kuesioner data

dikelompokkan menjadi *free format questions* dan *fixed format questions*.

Free format questions memberikan responden kebebasan menjawab. Jawaban responden pada kuesioner dengan *free format questions* sulit untuk ditabulasi dan ada kemungkinan jawaban tidak sesuai dengan pertanyaan.

Sedangkan *fixed format questions* memberikan responden pilihan jawaban atas pertanyaan yang diajukan. Hal ini membuat kuesioner dengan *fixed format questions* lebih mudah untuk ditabulasi.

2.1.8 Pemodelan Hubungan Entitas (*Entity Relationship Modeling*)

Menurut Connolly dan Begg (2010) *entity relationship modeling*, yang biasa disebut *ER modeling*, adalah rancangan basis data yang dibuat dengan menggunakan pendekatan *top-down*, yang dimulai dengan mengidentifikasikan data penting, yang disebut entitas (*entities*) dan relasi (*relationships*) antarentitas. Setelah itu, perlu pula ditambahkan informasi yang dimiliki oleh entitas dan relasi, yang disebut atribut (*attributes*) dan batasan (*constraints*) pada entitas, relationships, dan atribut.

ER modeling sangat berguna dalam perancangan basis data karena mampu menggambarkan fungsi dari data sehingga proses yang terjadi dalam sistem dapat dipahami dengan lebih baik. Konsep-konsep dasar dari *ER model* adalah entitas, relasi, atribut, dan batasan.

2.1.8.1 Tipe Entitas (*Entity Type*)

Mengacu pada pendapat Connolly dan Begg (2010) tipe entitas adalah sekumpulan objek dengan properti yang sama dimana objek tersebut didefinisikan oleh organisasi karena keberadaannya bebas (*independent existence*). *Entity occurrence* adalah suatu objek dari tipe entitas yang dapat diidentifikasikan secara unik.

Tiap tipe entitas digambarkan dengan menggunakan persegi panjang, yang di dalamnya bertuliskan nama dari entitas, yang biasanya berupa kata benda tunggal dan huruf awalnya menggunakan huruf kapital. Contoh representasi diagram dari tipe entitas terdapat pada Gambar 2.3 sebagai berikut:

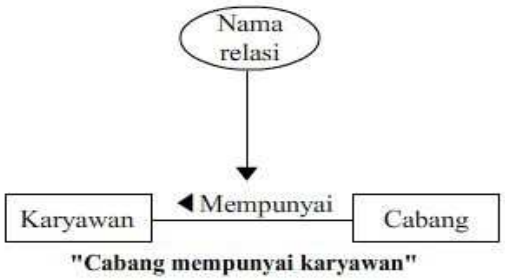
Gambar 2.3 Representasi Diagram dari Tipe Entitas Karyawan dan Cabang

2.1.8.2 Tipe Relasi (*Relationship Type*)

Menurut Connolly dan Begg (2010) tipe relasi adalah sekumpulan asosiasi antara satu atau lebih entitas yang saling terkait. Setiap tipe relasi diberikan sebuah nama untuk mendeskripsikan fungsinya masing-masing. *Relationship*

occurrence adalah suatu asosiasi yang dapat diidentifikasikan secara unik yang menggambarkan suatu kejadian (*occurence*) dari tiap tipe entitas yang berhubungan.

Tipe relasi digambarkan sebagai suatu garis yang menghubungkan tipe entitas yang saling terkait dan diberi label yang sesuai dengan nama dari tipe relasi tersebut. Pada umumnya nama dari tipe relasi menggunakan suatu kata kerja atau frase pendek yang di dalamnya terdapat kata kerja. Contoh representasi diagram dari tipe relasi terdapat pada Gambar 2.4 sebagai berikut:



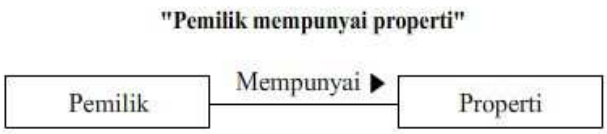
Gambar 2.4 Representasi Diagram dari Tipe Relasi Cabang Mempunyai Karyawan

2.1.8.2.1 *Degree of Relationship Type*

Berdasarkan pendapat Connolly dan Begg (2010) *degree of relationship type* adalah jumlah dari tipe entittas yang terlibat dalam suatu relasi. Jenis *degree of relationship type* antara lain:

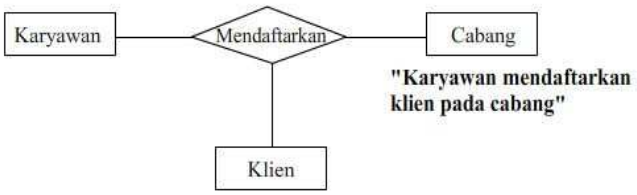
- *Binary*, yang merupakan *a relationship of degree two*. Pada tipe relasi *binary* terdapat dua entitas

yang saling berhubungan. Berikut ini adalah contoh relasi *binary* (Gambar 2.5):



Gambar 2.5 Relasi *Binary* dengan Nama Mempunyai

- *Ternary*, yang merupakan *a relationship of degree three*. Pada tipe relasi *ternary* terdapat tiga entitas yang saling berhubungan. Berikut ini adalah contoh relasi *ternary* (Gambar 2.6):



Gambar 2.6 Relasi *Ternary* dengan Nama Mendaftarkan

- *Quaternary*, yang merupakan *a relationship of degree four*. Pada relasi *quaternary* terdapat empat entitas yang saling berhubungan. Berikut ini adalah contoh relasi *quaternary* (Gambar 2.7):

Gambar 2.7 Relasi *Quaternary* dengan Nama Memberikan

2.1.8.2.2 Relasi Rekursif (*Recursive Relationship*)

Menurut Connolly dan Begg (2010) relasi rekursif adalah tipe relasi dimana tipe entitas yang sama memiliki lebih dari satu peran. Relasi rekursif sering disebut sebagai *unary relationship*. Berikut ini adalah contoh relasi rekursif (Gambar 2.8):

Gambar 2.8 Relasi Rekursif dengan Nama Mengawasi

2.1.8.3 Atribut (*Attribute*)

Mengacu pada pendapat Connolly dan Begg (2010) atribut adalah properti dari entitas atau tipe relasi. Atribut menyimpan

nilai yang menjelaskan tiap *entity occurrence* dan merepresentasikan data utama yang disimpan dalam basis data.

Menurut Connolly dan Begg (2010) *domain* atribut adalah sekumpulan nilai yang diperbolehkan untuk satu atau lebih atribut.

Atribut dapat diklasifikasikan menjadi:

2.1.8.3.1 *Simple* dan *Composite Attributes*

Simple attribute adalah atribut yang terdiri dari satu komponen tunggal, dengan keberadaan yang independen dan tidak dapat dibagi menjadi bagian yang lebih kecil. Contoh dari *simple attribute* adalah NIM, NIK, dan noKTP.

Composite attribute adalah atribut yang terdiri dari beberapa komponen. Contoh dari *composite attribute* adalah alamat, yang terdiri dari atribut jalan, kota, dan kodePos.

2.1.8.3.2 *Single-valued* dan *Multi-valued Attributes*

Single-valued attribute adalah atribut yang hanya memiliki nilai tunggal dari sebuah entitas. Contoh dari *single-valued attribute* adalah noCabang dan NIM.

Multi-valued attribute adalah atribut yang memiliki beberapa nilai dari sebuah entitas. Contoh dari *multi-valued attribute* adalah noTelp.

2.1.8.3.3 *Derived attributes*

Derived attribute adalah atribut yang nilainya dihasilkan dari satu atau beberapa atribut lainnya, namun tidak harus berasal dari entitas yang sama. Contoh dari *derived attribute* adalah lamaPinjam, yang nilainya diperoleh dari perhitungan tglPinjam dan tglKembali.

2.1.8.3.4 *Keys*

Menurut Connolly dan Begg (2010) *candidate key* adalah sekumpulan atribut dengan jumlah yang minimal, yang secara unik mengidentifikasikan tiap kejadian dari tipe entitas.

Berdasarkan pada pendapat Connolly dan Begg (2009) *primary key* adalah *candidate key* yang dipilih untuk mengidentifikasikan secara unik tiap kejadian dari tiap entitas.

Menurut Connolly dan Begg (2009) *composite key* adalah *candidate key* yang terdiri dari dua atau lebih atribut.

Mengacu pada pendapat Whitten, Bentley, dan Dittman (2004) *alternate key* adalah *candidate key* yang tidak terpilih menjadi *primary key*. *Alternate key* disebut juga dengan *secondary key*.

2.1.8.4 *Strong* dan *Weak Entity Types*

Menurut Connolly dan Begg (2010) tipe entitas diklasifikasikan menjadi:

a. *Strong entity type* adalah tipe entitas yang keberadaannya tidak bergantung pada entitas lainnya.
b. *Weak entity type* adalah tipe entitas yang keberadaannya bergantung pada entitas lainnya.

2.1.8.5 Atribut pada Relasi

Atribut dapat digunakan untuk memperjelas detail tentang relasi antarentitas. Untuk merepresentasikan atribut yang berhubungan dengan tipe relasi dapat dipergunakan simbol yang sama seperti simbol untuk menggambarkan tipe entitas. Contoh relasi dengan atributnya terdapat pada Gambar 2.9 berikut ini:

Gambar 2.9 Relasi Mengiklankan dengan Atribut tanggalIklan dan biaya

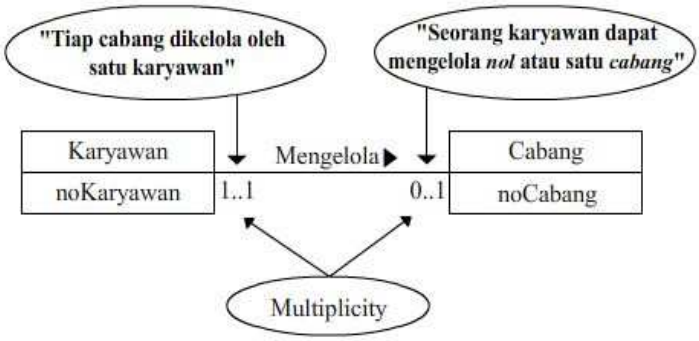
2.1.8.6 *Structural Constraints*

Batasan pada tipe entitas yang berpartisipasi dalam relasi harus merefleksikan batasan yang terjadi pada relasi di dunia nyata. Jenis batasan utama pada relasi biasa disebut dengan *multiplicity*. Menurut Connolly dan Begg (2010) *multiplicity* adalah jumlah (atau *range*) dari kejadian yang mungkin terjadi pada suatu tipe entitas yang terhubung dengan kejadian dari tipe entitas lainnya pada suatu relasi.

2.1.8.6.1 *Multiplicity for Binary Relationships*

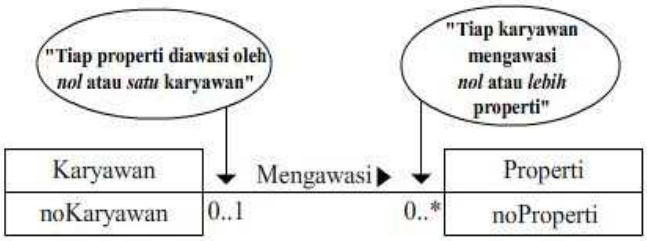
Pada umumnya derajat relasi yang sering terjadi adalah *binary*. Contoh *multiplicity* yang biasa terjadi pada relasi *binary* adalah:

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



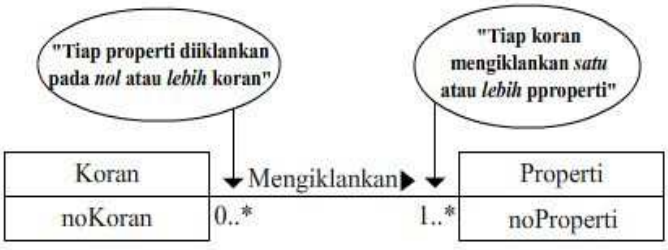
Gambar 2.10 *Multiplicity* dengan Relasi *One-to-One* (1:1)

b. *One-to-Many* (1:*) *Relationships*



Gambar 2.11 *Multiplicity* dengan Relasi *One-to-Many* (1:*)

c. *Many-to-Many* (*:*) *Relationships*



Gambar 2.12 *Multiplicity* dengan Relasi *Many-to-Many* (*:*)

2.1.8.6.2 *Multiplicity for Complex Relationships*

Multiplicity untuk *complex relationships* adalah jumlah (atau *range*) dari kejadian yang mungkin terjadi pada suatu tipe entitas dalam hubungan *n-ary* dimana nilai lainnya (n-1) bersifat tetap.

2.1.9 Metode Perancangan Basis Data (*Database Design Methodology*)

Menurut Connolly dan Begg (2010) metode perancangan basis data adalah pendekatan terstruktur yang menggunakan prosedur, teknik, *tools*, dan dokumentasi yang mendukung dan memfasilitasi proses perancangan basis data yang sesuai dengan kebutuhan perusahaan. Terdapat tiga fase dalam perancangan basis data antara lain:

2.1.9.1 Perancangan Basis Data Konseptual (*Conceptual Database Design*)

Berdasarkan pendapat Connolly dan Begg (2010) perancangan basis data konseptual berguna dalam membangun representasi konseptual dari basis data, yang meliputi identifikasi terhadap entitas, relasi, dan atribut yang penting.

- **Langkah 1: Membangun Model Data Konseptual**

  Langkah ini bertujuan untuk membangun model data konseptual yang mendukung kebutuhan (*data requirements*) perusahaan. Model data konseptual terdiri dari tipe entitas, tipe

relasi, atribut dan *domain* atribut, *primary key* dan *alternate keys*, serta *integrity constraints*. Untuk membangun model data konseptual diperlukan langkah-langkah:

a. Langkah 1.1: Mengidentifikasi tipe entitas

Langkah ini bertujuan untuk mengidentifikasi tipe entitas apa saja yang dibutuhkan dalam sistem. Tipe entitas dapat diperoleh dengan cara menentukan spesifikasi kebutuhan perusahaan (*users' requirements spesification*). Cara lain untuk mengidentifikasi entitas ialah dengan mencari objek-objek yang dapat berdiri sendiri.

b. Langkah 1.2: Mengidentifikasi tipe relasi

Langkah ini bertujuan untuk mengidentifikasi relasi penting yang terjadi antartipe entitas. Tipe relasi dapat diketahui dengan menggunakan tata bahasa (*grammar*) dari spesifikasi kebutuhan. Sebagai contoh, diketahui bahwa *Staff Manages PropertyForRent*. Hal ini menunjukkan bahwa hubungan antara entitas *Staff* dan *PropertyForRent* dibutuhkan di dalam sistem, dimana nama hubungannya ialah *Manages*.

c. Langkah 1.3: Mengidentifikasi dan menghubungkan atribut dengan tipe entitas atau relasi

Langkah ini bertujuan untuk menghubungkan atribut-atribut dengan tipe entitas atau relasi yang tepat. Atribut yang telah diidentifikasi perlu diberi nama. Setelah itu untuk tiap atribut perlu disertakan informasi mengenai:

- nama dan deskripsi atribut,
- tipe data dan panjangnya,
- *aliases* untuk atribut tersebut,
- apakah atribut bersifat *composite*, jika ya, atribut *simple* apa saja yang menyusunnya,
- apakah atribut bersifat *multi-valued*,
- apakah atribut bersifat *derived*, jika ya, bagaimana cara penghitungannya,
- *default value* untuk atribut.

d. Langkah 1.4: Menentukan *domain* atribut

Tujuan dari langkah ini ialah untuk menentukan *domain* untuk atribut yang terdapat pada model data konseptual. *Domain* adalah aturan untuk nilai yang dapat di-*input* ke dalam atribut.

e. Langkah 1.5: Menentukan atribut *candidate key*, *primary key*, dan *alternate key*

Pada langkah ini dilakukan pengidentifikasian *candidate key* untuk tiap tipe entitas dan, jika terdapat lebih dari satu *candidate key*, maka salah satunya dapat dipilih untuk menjadi *primary key* sedangkan yang lainnya menjadi *alternate keys*.

f. Langkah 1.6: Mempertimbangkan penggunaan *enhanced modeling concepts* (*optional*)

Langkah ini dapat dilakukan dan dapat pula tidak dilakukan karena bersifat *optional*. Tujuan dari langkah ini ialah untuk mempertimbangkan penggunaan konsep *enhanced modeling*, seperti spesialisasi/generalisasi, agregasi, dan komposisi, sebagai langkah lanjutan dalam pengembangan *ER model*.

g. Langkah 1.7: Memeriksa redudansi pada model

Pada langkah ini dilakukan pemeriksaan apakah redudansi atau pengulangan masih terjadi pada model data konseptual. Jika masih terdapat redudansi, maka redundansi tersebut harus dihilangkan.

h. Langkah 1.8: Memvalidasi model data konseptual terhadap transaksi pengguna

Langkah ini bertujuan untuk memastikan model data yang telah dibuat mampu mendukung transaksi-transaksi yang diperlukan.

i. Langkah 1.9: Meninjau kembali model data konseptual dengan pengguna

Tujuan dari langkah ini adalah untuk memeriksa kembali model data konseptual bersama dengan pengguna untuk memastikan bahwa model yang dibuat merupakan representasi sebenarnya dari apa yang dibutuhkan oleh perusahaan. Model data konseptual yang diperiksa mencakup *ER diagram* dan dokumentasi yang berhubungan dengan model data.

2.1.9.2 Perancangan Basis Data Logikal (*Logical Database Design*)

Mengacu pada pendapat Connolly dan Begg (2010) perancangan basis data merupakan proses pembuatan model data logikal yang berdasar pada model data konseptual, namun tidak bergantung pada DBMS tertentu dan pertimbangan fisikal lainnya.

- **Langkah 2: Membangun Model Data Logikal**

Langkah ini bertujuan untuk mengubah model data konseptual menjadi model data logikal dari basis data. Selain itu, perancangan basis data logikal juga memvalidasi model data logikal untuk memeriksa apakah struktur yang dibuat sudah benar dan mampu memenuhi kebutuhan transaksi perusahaan. Terdapat beberapa langkah yang perlu dilakukan untuk membangun model data logikal yaitu:

a. Langkah 2.1: Membangun relasi untuk model data logikal

Langkah ini bertujuan untuk membuat relasi untuk model data logikal, yang berguna untuk mewakili entitas, relasi, dan atribut yang telah diidentifikasi. Komposisi dari tiap relasi dijelaskan dengan menggunakan DBDL untuk basis data relasional.

b. Langkah 2.2: Memvalidasi relasi menggunakan normalisasi

Pada langkah ini dilakukan normalisasi yang berguna untuk memvalidasi relasi pada model data logikal. Tujuan dari tahap ini ialah untuk memastikan sekumpulan relasi memiliki jumlah atribut yang minimal, dimana atribut tersebut mampu memenuhi kebutuhan data perusahaan.

Selain itu, harus dipastikan pula bahwa redudansi yang terjadi pada relasi tidak ada atau seminimal mungkin.

c. Langkah 2.3: Memvalidasi relasi terhadap transaksi pengguna

Tujuan dari langkah ini ialah untuk memastikan relasi yang terdapat pada model data logikal mendukung transaksi-transaksi yang dibutuhkan. Selain itu, pada langkah ini juga dilakukan pemastian tidak adanya kesalahan (*error*) saat pembuatan relasi.

d. Langkah 2.4: Memeriksa batasan integritas (*integrity constraints*)

Langkah ini ditujukan untuk memeriksa apakah batasan integritas telah diterapkan dengan benar pada model data logikal. Batasan integritas berguna untuk melindungi basis data dari ketidaklengkapan, ketidakakuratan, dan ketidakkonsistenan.

e. Langkah 2.5: Memeriksa kembali model data logikal bersama pengguna

Tujuan dilakukannya langkah ini ialah untuk melakukan pemeriksaan kembali model data logikal bersama dengan pengguna guna memastikan bahwa model

yang dibuat telah sesuai dengan representasi kebutuhan data perusahaan.

f. Langkah 2.6: Menggabungkan model data logikal ke dalam model data global (*optional*)

Langkah ini bertujuan untuk menggabungkan model data logikal lokal menjadi model data logikal global, yang merepresentasikan semua sudut pandang pengguna dari sebuah basis data. Langkah ini bersifat *optional* karena hanya dibutuhkan oleh basis data dengan banyak sudut pandang.

g. Langkah 2.7: Mempertimbangkan perkembangan di masa depan

Langkah ini bertujuan untuk menentukan apakah akan muncul perubahan yang signifikan pada masa yang akan datang dan memastikan apakah model data logikal mampu mengakomodasi perubahan tersebut.

2.1.9.3 Perancangan Basis Data Fisikal (*Physical Database Design*)

Menurut Connolly dan Begg (2010) perancangan basis data fisikal adalah proses menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder (*secondary storage*), dimana deskripsi basis data fisikal tersebut menggambarkan relasi dasar

(*base relations*), organisasi *file* (*file organizations*), dan indeks (*indexes*) yang berguna untuk memperoleh akses data dengan efisien, dan batasan integritas (*integrity constraints*) serta pengukuran keamanan (*security measures*) terkait.

Berdasarkan pendapat Connolly dan Begg (2010) terdapat beberapa metodologi perancangan basis data fisikal, yaitu:

- **Langkah 3: Menerjemahkan Model Data Logikal untuk DBMS yang telah ditentukan**

  Tujuan dari langkah ini ialah untuk menghasilkan skema basis data relasional dari model data logikal yang dapat diimplementasikan pada DBMS tujuan. Langkah-langkah yang perlu dilakukan agar dapat menghasilkan skema basis data relasional tersebut terdiri dari:

  a. Langkah 3.1: Merancang relasi dasar

  Tujuan dari langkah ini adalah untuk menentukan bagaimana cara merepresentasikan relasi dasar, yang telah diidentifikasi pada model data logikal, ke dalam DBMS tujuan.

  b. Langkah 3.2: Merancang representasi *derived data*

  Pada langkah ini dilakukan penentuan bagaimana perepresentasian *derived data*, yang terdapat pada model

data logikal, ke dalam DBMS tujuan. *Derived* atau *calculated attribute* adalah atribut yang nilainya dapat diperoleh dari nilai atribut-atribut lainnya.

c. **Langkah 3.3: Merancang batasan umum (*general constraints*)**

Tujuan dari langkah ini adalah untuk merancang batasan umum untuk DBMS tujuan. Proses *update* pada relasi dapat dibatasi oleh batasan integritas yang mengatur transaksi sebenarnya.

- **Langkah 4: Merancang organisasi *file* (*file organtization*) dan indeks**

Pada langkah ini ditentukan organisasi file yang optimal yang berguna untuk menyimpan relasi dasar dan indeks, yang dibutuhkan untuk mendapatkan kinerja yang baik, yaitu dengan cara penyimpanan relasi dan *tuples* pada penyimpanan sekunder. Langkah ini terdiri dari:

a. **Langkah 4.1: Menganalisis transaksi**

Tujuan dari langkah ini ialah untuk memahami fungsionalitas transaksi yang berjalan pada basis data dan untuk menganalisis transaksi-transaksi penting.

b. Langkah 4.2: Memilih organisasi *file*

Pada langkah ini dilakukan penentuan organisasi *file* yang efisien pada tiap relasi dasar. Salah satu tujuan utama dari perancangan basis data fisikal adalah untuk menyimpan dan mengakses data secara efisien.

c. Langkah 4.3: Memilih indeks

Tujuan dari dilakukannya langkah ini adalah untuk menentukan apakah penambahan indeks akan meningkatkan kinerja sistem.

d. Langkah 4.4: Memperkirakan kebutuhan kapasitas *disk* (*disk space*)

Memperkirakan kapasitas *disk* yang dibutuhkan oleh basis data sangat bergantung pada DBMS tujuan dan perangkat keras yang digunakan untuk mendukung basis data.

- **Langkah 5: Merancang *User Views***

Langkah ini bertujuan untuk merancang *user views* yang diidentifikasi selama tahap pengumpulan dan penganalisisan kebutuhan dari siklus hidup sistem basis data.

- **Langkah 6: Merancang Mekanisme Keamanan**

  Tujuan dari langkah ini adalah untuk merancang mekanisme keamanan untuk basis data untuk mencegah kehilangan atau penyalahgunaan data.

- **Langkah 7: Mempertimbangkan Penggunaan Redudansi Terkontrol**

  Langkah ini bertujuan untuk menentukan apakah penggunaan redudansi secara terkontrol, dengan tidak terlalu berpatokan pada aturan normalisasi, akan meningkatkan kinerja sistem. Terdapat beberapa langkah yang perlu dilakukan, antara lain:

  a. **Langkah 7.1: Menggabungkan relasi *one-to-one* (1:1)**

     Langkah ini bertujuan untuk menentukan efek dari penggabungan relasi ke dalam sebuah relasi tunggal.

  b. **Langkah 7.2: Menduplikasi *non-key attributes* pada relasi *one-to-many* (1:*) untuk mengurangi *joins***

     Dengan tujuan khusus untuk mengurangi atau meniadakan *joins* dari *queries* yang penting atau sering digunakan, akan menghasilkan keuntungan-keuntungan yang berdampak pada pengulangan satu atau lebih *non-key attributes*.

c. Langkah 7.3: Menduplikasi atribut *foreign key* pada relasi *one-to-many* (1:*) untuk mengurangi *joins*

Dengan memperbanyak atribut *foreign key* pada relasi 1:*, *joins* akan dapat dikurangi bahkan dihilangkan. Menghilangkan dua *joins* dari *queries* akan menciptakan relasi secara langsung antara dua entitas, sehingga *foreign key* pada salah satu entitas harus diperbanyak.

d. Langkah 7.4: Menduplikasi atribut pada relasi *many-to-many* (*:*) untuk mengurangi *joins*

Pada perancangan basis data logikal, relasi *:* dipisahkan menjadi tiga relasi: dua relasi diturunkan dari entitas asli dan satu relasi baru menampilkan relasi antara dua entitas. Jika ingin menghasilkan informasi dari relasi *:*, maka tiga relasi tersebut harus disatukan.

e. Langkah 7.5: Memperkenalkan kelompok yang berulang (*repeating groups*)

Kelompok data yang berulang telah dihilangkan dari model data logikal sebagai akibat dari normalisasi 1NF. Kelompok data berulang dipisahkan ke dalam relasi baru, yang membentuk relasi 1:* dengan relasi asalnya (*parent*).. Biasanya kelompok data berulang hanya menyimpan nilai

saat ini, atau ada pula yang menyimpan nilai yang sering digunakan.

f. Langkah 7.6: Membuat *extract tables*

Dengan adanya *extract tables*, waktu yang diperlukan untuk mengakses sistem akan lebih hemat karena pengguna diperbolehkan untuk mengakses *extract tables* secara langsung, daripada mengakses relasi dasar.

g. Langkah 7.7: Memisahkan (*partitioning*) relasi

Partition adalah suatu pendekatan yang membagi relasi dan indeks dengan ukuran yang besar ke dalam bagian-bagian yang lebih kecil dan lebih mudah dikelola.

- **Langkah 8: Memonitor dan Merancang Sistem Operasi**

Tujuan dari langkah ini ialah untuk memonitor atau mengawasi sistem operasi dan meningkatkan kinerja sistem untuk memperbaiki keputusan rancangan yang tidak sesuai, atau menerapkan perubahan kebutuhan.

2.1.10 Normalisasi (*Normalization*)

Mengacu pada pendapat Connolly dan Begg (2010) normalisasi adalah suatu teknik untuk menghasilkan sekumpulan relasi dengan properti

yang diinginkan, yang sesuai dengan kebutuhan data (*data requirements*) perusahaan.

2.1.10.1 Tujuan Normalisasi

Menurut Connolly dan Begg (2010) tujuan utama dari normalisasi ialah untuk mengidentifikasi sekumpulan relasi yang dapat mendukung kebutuhan data perusahaan atau organisasi. Hasil dari normalisasi berupa entitas-entitas yang disimpan dalam basis data sehingga tidak ada lagi data yang berulang (redudansi data). Berikut ini adalah karakteristik sekumpulan relasi yang sesuai:

- jumlah atribut harus minimal untuk memenuhi kebutuhan transaksi dalam perusahaan,
- atribut dengan *logical relationship* yang dekat (biasa disebut *functional dependency*) harus dikelompokkan ke dalam relasi yang sama,
- redudansi yang dihasilkan harus minimal (tiap atribut hanya muncul sekali)

2.1.10.2 Proses Normalisasi

Pada pembuatan skripsi ini proses normalisasi yang digunakan terdiri dari tahap 1NF, 2NF, dan 3NF.

2.1.10.2.1 *First Normal Form* (1NF)

Menurut Connolly dan Begg (2010) proses normalisasi dimulai dengan cara memindahkan data dari sumber (contohnya dari *form entry data*) ke dalam format tabel yang memiliki baris dan kolom. Pada bentuk ini tabel memuat bentuk yang masih belum normal (*unnormalized form* - UNF). Tabel yang belum normal ini dikatakan *unnormalized table* karena tabel masih memuat satu atau lebih data yang berulang.

Setelah terbentuk *unnormalized table*, selanjutnya data dipindahkan ke dalam bentuk 1NF, dengan cara mengidentifikasi dan menghilangkan data yang berulang. Terdapat dua pendekatan untuk menghilangkan data yang berulang dari *unnormalized table*:

a. Dengan memasukkan data yang sesuai pada kolom kosong yang terdapat pada baris yang memuat data berulang. Pendekatan ini biasa disebut "*flattening the table*" karena kolom kosong perlu diisi dengan data yang tidak berulang, ketika diperlukan.
b. Dengan menempatkan data berulang, bersama dengan salinan atribut kunci yang asli (*original key attributes*), pada relasi yang terpisah. Pendekatan

ini terus dilakukan sampai tidak ada lagi data berulang dalam tabel.

2.1.10.2.2 *Second Normal Form* (2NF)

Menurut Connolly dan Begg (2010) 2NF ditandai dengan terbentuknya relasi pada 1NF dan tiap atribut yang bukan *primary key* bergantung secara penuh (*fully functionally dependent*) pada *primary key*. Pada tahap 2NF dilakukan penghilangan *partial dependencies*.

Fully functional dependency terjadi jika A dan B adalah atribut dari relasi, B bergantung penuh pada A (A→B) jika B *functionally dependent* pada A, namun bukan merupakan bagian dari A. Sedangkan *functional dependency* A→B disebut *partial dependency* jika terdapat beberapa atribut yang dapat dihapus dari A, namun ketergantungan masih tetap berlanjut.

2.1.10.2.3 *Third Normal Form* (3NF)

Mengacu pada pendapat Connolly dan Begg (2010) 3NF ditandai dengan terbentuknya relasi pada 1NF dan 2NF dan tidak ada atribut yang bukan *primary key* yang bergantung secara transitif

(*transitive dependent*) pada *primary key*. Untuk mendapatkan relasi pada 3NF perlu dilakukan penghilangan *transitive dependent* yang terdapat pada relasi yang dihasilkan dari tahap 2NF.

Transitive dependency adalah suatu kondisi dimana A, B, dan C merupakan atribut dari relasi A→B dan B→C. C dikatakan memiliki hubungan transitif dengan A melalui B (A tidak *functionally dependent* dengan B dan C).

2.2 Pemahaman Studi yang Berhubungan dengan Klinik Gigi

Didalam pemahaman studi ini menjelaskan tentang seluruh komponen yan berhubungan dengan Klinik Gigi, yaitu:

2.2.1 Klinik Gigi

Menurut Sharda, Srinath, Ramesh, Nagesh, dan Kailash (2011: 16) dewasa ini kesehatan gigi dan mulut (*oral health*) disadari sama pentingnya dengan kesehatan umum (*general health*). Masalah dalam kesehatan gigi dan mulut dapat menyebabkan timbulnya rasa sakit dan kehilangan gigi. Di samping itu, penyakit gigi dan mulut juga dapat mempengaruhi penampilan, kualitas hidup, dan asupan gizi seseorang.

Kesehatan gigi dan mulut sangat erat kaitannya dengan kesadaran dan pengetahuan,akan pentingnya menjaga kesehatan gigi dan mulut, serta

kebiasaan hidup seseorang. Untuk terus menjaga kesehatan gigi dan mulut, pemeriksaan ke klinik gigi dapat terus dilakukan secara berkala.

Mengacu pada Peraturan Menteri Kesehatan Republik Indonesia No. 028/MENKES/PER/I/2011 Tentang Klinik, klinik adalah fasilitas penyedia pelayanan kesehatan perorangan, yang memberikan pelayanan medis baik dasar maupun spesialistik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan, dengan dipimpin oleh seorang tenaga medis. Pada klinik gigi tenaga medis yang memimpin adalah dokter gigi atau dokter gigi spesialis.

Klinik merupakan salah satu bentuk fasilitas yang menyediakan pelayanan kesehatan yang dibutuhkan oleh masyarakat, guna terselenggaranya pelayanan kesehatan yang mudah diakses, terjangkau, dan bermutu. Menurut jenis pelayanan yang diberikan, klinik dibedakan menjadi:

- Klinik Pratama

  Klinik Pratama adalah klinik yang dipimpin oleh seorang dokter atau dokter gigi. Berdasarkan fungsinya, Klinik Pratama harus mampu memberikan pelayanan medik dasar. Tenaga medis yang harus dimiliki oleh sebuah Klinik Pratama minimal berjumlah dua dokter.

- Klinik Utama

  Klinik Utama adalah klinik yang memberikan pelayanan medik spesialistik atau pelayanan medik dasar dan spesialistik. Klinik Utama dipimpin oleh dokter spesialis atau dokter gigi spesialis. Pada Klinik Utama tenaga medis yang diperlukan minimal berjumlah satu dokter

spesialis dari masing-masing spesialisasi yang diberikan. Di samping itu, diperlukan pula dokter atau dokter gigi yang berguna sebagai tenaga pelaksana pelayanan medis.

2.2.1.1 Kewajiban Klinik

Menurut Peraturan Menteri Kesehatan Republik Indonesia No. 028/MENKES/PER/I/2011 Tentang Klinik, kewajiban yang perlu diemban oleh klinik dalam memberikan pelayanan terdiri dari:

a. memberikan pelayanan yang aman dan bermutu, dengan mengutamakan kepentingan pasien sesuai dengan standar profesi, standar pelayanan, serta standar prosedur opersional,
b. memberikan pelayanan gawat darurat kepada pasien, tanpa mendahulukan kepentingan finansial,
c. memperoleh persetujuan atas tindakan yang akan dilakukan,
d. menyelenggarakan rekam medis,
e. melaksanakan sistem rujukan,
f. menolak keinginan pasein yang bertentangan dengan standar profesi, etika, dan peraturan perundang-undangan,
g. menghormati hak pasien,
h. melaksanakan kendali mutu dan kendali biaya,
i. memiliki peraturan internal dan standar prosedur operasional,
j. melaksanakan program pemerintah dalam bidang kesehatan, baik secara regional maupun nasional.

2.2.1.2 Pelayanan

Berdasarkan Hasil Riset Kesehatan Dasar (RISKESDAS) 2007 yang diselenggarakan oleh Kementerian Kesehatan RI, rata-rata masyarakat Indonesia memiliki kurang lebih 5 gigi rusak untuk tiap orangnya. Dari gigi yang rusak tersebut, yang telah ditambal hanya 0.7%. (diperoleh dari http://www.pdgi.or.id/news/detail/gigi-dan-mulut-sehat-untuk-kualitas-hidup-yang-lebih-baik-pada-hari-kesehatan-gigi-2012, diakses pada 18 Oktober 2012).

Hasil survei tersebut menunjukkan bahwa tingkat kesehatan gigi masyarakat Indonesia masih rendah. Padahal kualitas kesehatan gigi yang baik akan meningkatkan kualitas hidup seseorang.

Berdasarkan Keputusan Menteri Kesehatan RI No. 039/MENKES/SK/I/2007 Tentang Pedoman Penyelenggaraan Kedokteran Gigi Keluarga, pelayanan kedokteran gigi keluarga dilaksanakan dengan berpedoman pada pola pelayanan berlapis, melalui sistem rujukan berjenjang (*Level of Care*), dengan pendekatan *Primary Health Care* (PHC). Pendekatan PHC dapat dilakukan di puskesmas. rumah sakit, serta klinik pribadi.

Berikut ini adalah ruang lingkup pelayanan yang diberikan oleh dokter gigi:

1. Pelayanan Darurat (*Basic Emergency Care*)

   Pelayanan ini dilakukan dengan memberikan pertolongan pertama pada keadaan darurat. Apabila diperlukan, rujukan dapat diberikan kepada pasien.

2. Pelayanan Pencegahan (*Preventive Care*)

   Pelayanan pencegahan dilakukan dengan memberikan pendidikan kesehatan gigi kepada masyarakat. Pelayanan ini disebut juga sebagai tindakan penganan dini (*early detection and prompt treatment*).

3. Pelayanan Medik Gigi Dasar (*Simple care*)

   Pelayanan medik gigi dasar adalah pelayanan profesional sederhana, seperti pembersihan karang gigi, tumpatan gigi, ekstraksi gigi, dan pengelolaan halitosis.

4. Pelayanan Medik Gigi Khusus (*Moderate care*)

   Pelayanan medik gigi khusus adalah pelayanan professional di bidang kedokteran, seperti konservasi gigi, bedah mulut, dan *oral medicine*. Pelayanan ini hanya bisa dilakukan di rumah sakit dan oleh dokter gigi spesialis.

2.2.1.3 Pasien

Menurut Peraturan Menteri Kesehatan RI No. 269/MENKES/PER/III/2008 Tentang Rekam Medis, pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya guna memperoleh pelayanan kesehatan yang diperlukan, baik secara langsung maupun tidak langsung, kepada dokter atau dokter gigi.

2.2.1.4 Rekam Medis (*Medical Record*)

Menurut Peraturan Menteri Kesehatan RI No. 269/MENKES/PER/III/2008 Tentang Rekam Medis, yang dimaksud rekam medis adalah berkas yang berisi catatan dan dokumen berkaitan dengan identitas pasien, hasil pemeriksaan, pengobatan, serta tindakan lain yang diberikan kepada pasien. Rekam medis berperan penting dalam pelayanan pasien karena kelengkapan data dalam rekam medis mendukung pengambilan keputusan pengobatan, penanganan, dan tindakan medis.

Berdasarkan bentuknya, rekam medis dibedakan menjadi dua, yaitu:

1. Rekam medis konvensional

   Rekam medis konvensional yaitu rekam medis yang berbentuk lembaran kertas medis.

2. Rekam medis elektronik

   Rekam medis elektronik yaitu rekam medis yang berbentuk elektronik, dimana datanya tersimpan dalam suatu media komputer.

2.2.1.5 Obat

Menurut Peraturan Menteri Kesehatan No. 949/MENKES/PER/VI/2000 Tentang Registrasi Obat Jadi, obat jadi adalah sediaan atau paduan bahan, termasuk produk biologi dan kontrasepsi, yang siap digunakan untuk mempengaruhi atau

menyelidiki sistem fisiologi atau keadaan patologi dalam rangka penetapan diagnosa, pencegahan dan penyembuhan penyakit, serta pemulihan dan peningkatan kesehatan.

Pengertian obat secara umum adalah bahan atau zat yang berasal dari tumbuhan, hewan, mineral maupun zat kimia tertentu yang dapat digunakan untuk mengurangi rasa sakit, memperlambat proses penyakit dan atau menyembuhkan penyakit.

2.2.2 Rekam Medis Elektronik (*Electronic Medical Record*)

Menurut McGrath, Arar, dan Pugh (2007: 2), rekam medis elektronik merupakan penggunaan sistem komputer untuk menyimpan, mengorganisir, dan memperoleh informasi tentang pasien, yang berguna untuk meningkatkan kualitas dari pelayanan kesehatan. Rekam medis elektronik sangat bermanfaat karena dengan menggunakannya akurasi dan kelengkapan data medis dapat ditingkatkan. Selain itu, rekam medis elektronik lebih mudah dimengerti dan dibaca dari pada rekam medis yang menggunakan media kertas.

Menurut Smelcher, Miller-Jacobs, dan Kantrovich (2009: 83) hal-hal yang diperlukan agar dapat merancang sistem rekam medis elektronik yang mudah dimengerti dan digunakan terdiri dari:

- gunakan skema navigasi yang fleksibel untuk mengakomodasi berbagai spesialis, lingkungan, anggota tim medis, dan praktik individu,
- gunakan *default* yang sesuai agar beban kerja yang non-produktif dapat dikurangi,

- mungkinkan *rapid switching* antara pasien, yang konsisten dan sesuai dengan cara dokter biasa bekerja,
- mungkinkan pelimpahan hak akses dari pihak yang bertanggung jawab kepada anggota lain dari tim medis,
- sediakan beragam metode *data entry* yang dapat dipilih oleh dokter agar dapat bekerja dengan lebih leluasa,
- pertimbangkan penggunaan teknik visualisasi, integrasi, dan manipulasi yang inovatif guna meminimalkan beban kerja tenaga medis.

2.2.3 Penjualan

Berikut ini adalah fungsi, jaringan prosedur, dan dokumen yang terkait dengan transaksi penjualan:

2.2.3.1 Fungsi yang Terkait dalam Penjualan

Menurut Mulyadi (2001) penjualan adalah transaksi yang menjual barang atau jasa, yang dapat dilakukan secara tunai maupun kredit. Dalam penjualan secara tunai, perusahaan akan menyerahkan barang dan jasa setelah pembeli memberikan sejumlah kas kepada perusahaan. Sedangkan dalam penjualan secara kredit, perusahaan memiliki piutang kepada pelanggannya jika pesanan pelanggan telah dipenuhi. Adapun fungsi-fungsi terkait dalam penjualan secara kredit adalah:

1. Fungsi Penjualan

   Fungsi penjualan bertanggung jawab untuk menerima surat pesanan, meng-*edit* pesanan untuk menambahkan informasi yang belum ada pada surat pesanan tersebut, meminta otorisasi kredit, menentukan tanggal pengiriman dan dari gudang mana barang akan dikirim, serta mengisi surat pesanan pengiriman.

2. Fungsi Kredit

   Fungsi kredit bertanggung jawab penuh untuk meneliti status kredit pelanggan dan memberikan otorisasi kredit kepada pelanggan.

3. Fungsi Gudang

   Fungsi gudang bertanggung jawab untuk menyimpan dan menyiapkan barang yang dipesan dan menyerahkan barang ke fungsi pengiriman.

4. Fungsi Pengiriman

   Fungsi pengiriman bertanggung jawab untuk menyerahkan barang berdasarkan surat pesanan pengiriman yang diterimanya dari fungsi penjualan.

5. Fungsi Penagihan

   Fungsi penagihan bertanggung jawab untuk membuat dan mengirimkan faktur penjualan kepada pelanggan, serta membuat *copy* faktur bagi kepentingan pencatatan transaksi penjualan oleh fungsi akuntansi.

6. Fungsi Akuntansi

   Fungsi akuntansi bertanggung jawab untuk mencatat piutang dari seluruh transaksi penjualan dan mengirimkan pernyataan piutang kepada debitur, serta membuat laporan penjualan.

2.2.3.2 Jaringan Prosedur Pembentuk Sistem Penjualan

Menurut Mulyadi (2001) jaringan prosedur pembentuk sistem penjualan kredit yaitu:

1. Prosedur Order Penjualan

   Dalam prosedur ini, fungsi penjualan menerima pesanan dari pembeli dan menambahkan informasi penting pada surat pesanan dari pembeli. Setelah itu, fungsi penjualan membuat surat pesanan pengiriman dan mengirimkannya kepada fungsi-fungsi lainnya.

2. Prosedur Persetujuan Kredit

   Dalam prosedur ini, fungsi penjualan meminta persetujuan penjualan kredit kepada pembeli dari fungsi kredit.

3. Prosedur Pengiriman

   Dalam prosedur ini, fungsi pengiriman mengirimkan barang kepada pembelu sesuai dengan surat pesanan pengiriman.

4. Prosedur Penagihan

   Dalam prosedur ini, fungsi penagihan membuat faktur penjualan, kemudian mengirimkannya kepada pembeli.

5. Prosedur Pencatatan Piutang

   Dalam prosedur ini, fungsi akuntansi mencatat tembusan faktur penjualan ke dalam kartu piutang.

6. Prosedur Distribusi Penjualan

   Dalam prosedur ini, fungsi akuntansi mendistribusikan data penjualan sesuai dengan informasi yang diperlukan oleh manajemen.

7. Prosedur Pencatatan Harga Pokok Penjualan

   Dalam prosedur ini, fungsi akuntansi mencatat secara periodik total harga pokok produk yang dijual dalam periode tertentu.

2.2.3.3 Dokumen yang Terkait dalam Prosedur Penjualan

Menurut Mulyadi (2001) dokumen yang digunakan dalam sistem penjualan kredit terdiri dari:

1. Surat pesanan pengiriman dan tembusannya
2. Faktur dan tembusannya
3. Rekapitulasi harga pokok penjualan
4. Bukti memorial

2.2.4 Pembelian

Berikut ini adalah fungsi, jaringan prosedur, dan dokumen yang terkait dengan transaksi pembelian:

2.2.4.1 Fungsi yang Terkait dalam Pembelian

Menurut Mulyadi (2001) pembelian adalah suatu transaksi yang dilakukan oleh suatu perusahaan untuk mengadakan barang-barang yang dibutuhkan oleh perusahaan. Transaksi pembelian dibedakan menjadi pembelian lokal dan impor. Pembelian lokal adalah pembelian yang dilakukan dari pemasok dalam negeri. Sedangkan pembelian impor adalah pembelian yang dilakukan dari pemasok luar negeri.

Adapun fungsi-fungsi yang terkait dalam sistem pembelian, yaitu:

1. Fungsi Gudang

   Dalam sistem pembelian, fungsi gudang bertanggung jawab untuk mengajukan permintaan pembelian sesuai dengan persediaan yang ada di gudang dan untuk menyimpan barang yang telah diterima oleh fungsi penerimaan.

2. Fungsi Pembelian

   Fungsi pembelian bertanggung jawab untuk mendapatkan informasi harga barang, menentukan pemasok dalam pengadaan barang, dan mengeluarkan surat pesanan pembelian barang kepada pemasok yang dipilih.

3. Fungsi Penerimaan

   Fungsi penerimaan bertanggung jawab untuk melakukan pemeriksaan terhadap jenis, kualitas, dan kuantitas barang yang diterima dari pemasok, serta menerima barang dari pembeli yang berasal dari transaksi retur penjualan.

4. Fungsi Akuntansi

   Fungsi akuntansi yang terkait dalam transaksi pembelian adalah fungsi untuk mencatat seluruh utang dan fungsi pencatatan persediaan. Dalam sistem pembelian, fungsi pencatat utang bertanggung jawab untuk mencatat seluruh transaksi pembelian dan membuat arsip dokumen sumber (bukti kas keluat) yang berfungsi sebagai catatan utang.

2.2.4.2 Jaringan Prosedur Pembentuk Sistem Pembelian

Dalam melakukan prosedur pembelian perlu dilakukan jaringan prosedur agar dapat terjadinya transaksi pembelian. Menurut Mulyadi (2001) jaringan prosedur yang membentuk sistem pembelian terdiri dari:

1. Prosedur Permintaan Pembelian

   Dalam prosedur ini fungsi gudang mengajukan permintaan pembelian dalam bentuk formulir surat permintaan pembelian kepada fungsi pembelian.

2. Prosedur Permintaan Penawaran Harga dan Pemilihan Pemasok

   Dalam prosedur ini fungsi pembelian mengirimkan surat permintaan penawaran harga kepada pemasok yang telah bekerja sama guna memperoleh informasi mengenai harga barang dan syarat pembelian lainnya.

3. Prosedur Pesanan Pembelian

   Dalam prosedur ini fungsi pembelian mengirim surat pesanan pembelian kepada pemasok yang dipilih dan memberitahukannya kepada unit-unt organisasi lain dalam perusahaan tentang order pembelian yang telah dikeluarkan.

4. Prosedur Penerimaan Barang

   Dalam prosedur ini fungsi penerimaan memeriksa jenis, kuantitas, dan kualitas barang dari pemasok, kemudian membuat laporan penerimaan barang untuk menyatakan penerimaan dari pemasok tersebut.

5. Prosedur Pencatatan Utang

   Dalam prosedur ini fungsi akuntansi memeriksa dokumen-dokumen yang berhubungan dengan pembelian dan melakukan pencatatan utang, atau mengarsipkan dokumen sumber sebagai catatan utang.

6. Prosedur Distribusi Pembelian

   Prosedur ini meliputi distribusi rekening yang di-debit dari transaksi pembelian guna mendukung pembuatan laporan manajemen.

2.2.4.3 Dokumen yang Terkait dalam Prosedur Pembelian

Menurut Mulyadi (2001), dokumen yang digunakan dalam sistem pembelian terdiri dari:

1. Surat Permintaan Pembelian
2. Surat Permintaan Penawaran Harga
3. Surat Pesanan Pembelian
4. Laporan Penerimaan Barang
5. Surat Perubahan Pesanan
6. Bukti Kas Keluar

2.3 Interaksi Manusia dan Komputer (IMK)

Menurut Shneiderman dan Plaisant (2010) terdapat 8 aturan emas perancangan antar muka (user interface), atau yang kerap disebut dengan 8 golden rules, yaitu:

1. Berusaha untuk konsisten (*strive for consistency*)

   Konsistensi diperlukan dalam urutan tindakan, istilah dalam *prompt*, menu, *help screens*, serta penggunaan warna, *layout*, kapitalisasi, jenis huruf, dan sebagainya.

2. Memungkinkan pengguna untuk menggunakan *shortcut* (*enable frequent users to use shortcuts*)

   Untuk meningkatkan kecepatan interaksi dapat digunakan singkatan, *special keys*, *hidden commands*, dan fasilitas makro.

3. Memberikan umpan balik yang informative (*offer informative feedback*)

   Sistem harus selalu memberikan respon, sebagai umpan balik, dari tindakan pengguna. Jika tindakan sering dilakukan dan tidak terlalu penting, maka respon yang diberikan cukup sederhana saja. Sedangkan jika tindakan jarang dilakukan dan penting, maka perlu diberikan respon yang *substantial*.

4. Merancang dialog untuk memberikan penutupan (*design dialog to yield closure*)

   Tindakan perlu diorganisir ke dalam kelompok dengan bagian awal, tengah, dan akhir. Umpan balik yang informatif yang muncul pada akhir dari serangkaian tindakan akan member tahu pengguna bahwa tindakan mereka sudah berhasil dilakukan.

5. Memberikan penanganan kesalahan yang sederhana (*offer simple error handling*)

   Sedapat mungkin sistem yang dirancang tidak memungkinkan pengguna untuk melakukan kesalahan yang fatal. Apabila suatu kesalahan terjadi, sistem harus dapat mendeteksi dan memberikan mekanisme penanganan kesalahan yang sederhana dan mudah dipahami.

6. Mudah kembali ke tindakan sebelumnya (*permit easy reversal of actions*)

   Fitur ini berguna untuk mengurangi kekhawatiran pengguna karena pengguna tahu bahwa kesalahan yang dilakukan dapat dibatalkan, sehingga pengguna tidak ragu untuk mengeksplorasi pilihan-pilihan yang belum ia kenal sebelumnya.

7. Mendukung tempat pengendali internal (*support internal locus of control*)

   Pengguna yang berpengalaman memiliki keinginan agar dapat menjadi pengendali sistem dan sistem dapat terus merespon tindakan yang dilakukannya. Sistem yang dirancang harus mampu membuat pengguna merasa bahwa ia adalah pengendali sistem, bukan responden.

8. Mengurangi beban ingatan jangka pendek (*reduce short-term memory load*)

   Beban ingatan jangka pendek dapat dikurangi dengan cara merancang layar sederhana dimana pilihan yang disajikan dapat dilihat dengan jelas, frekuensi *window-motion* dikurangi, dan waktu pelatihan untuk *codes, mnemonics*, urutan tindakan yang memadai.

2.4 *Tools* yang Digunakan

Tools yang digunakan dalam pembuatan skripsi ini antara lain:

2.4.1 *Tools Diagram*

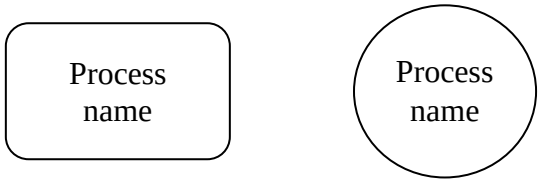
Tools diagram yang digunakan ialah Diagram Aliran Data, *State Transition Diagram*, dan *flowchart*.

2.4.1.1 Diagram Aliran Data (*Data Flow Diagram*)

Menurut Whitten, Bentley, dan Dittman (2004), *data flow diagram* adalah *process model* yang menggambarkan aliran data yang ada dalam sistem dan pemrosesan yang dilakukan oleh sistem. Dalam membuat DFD diperlukan beberapa simbol, yang terdiri dari:

1. Proses (*Process*)

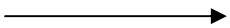
   Proses merupakan pekerjaan yang dilakukan oleh sistem sebagai respon dari aliran data yang masuk.



Gambar 2.13 Bentuk Proses (kiri - Gane and Sarson shape, kanan - DeMarco/Yourdon shape)

2. Aliran Data (*Data Flow*)

   Aliran data merupakan data yang menjadi masukan bagi atau keluaran dari suatu proses.



Gambar 2.14 Bentuk Aliran Data

3. *External Agent*

   Merupakan satuan di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang berinteraksi dengan sistem.

Gambar 2.15 Bentuk *External Agent*

(kiri - Gane and Sarson shape, kanan - DeMarco/Yourdon shape)

4. *Data Store*

Data Store merupakan simbol yang digunakan untuk menunjukkan suatu tempat penyimpanan data yang nantinya akan digunakan.

Gambar 2.16 Bentuk *Data Store*

(kiri - Gane and Sarson shape, kanan - DeMarco/Yourdon shape)

Jenis-jenis *Data Flow Diagram* (DFD) adalah sebagai berikut:

1. Diagram Konteks

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.

2. Diagram Nol (*Overview Diagram*)

Diagram nol adalah diagram yang menggambarkan proses dari *data flow diagram*. Diagram nol memberikan gambaran secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data dan *external entity*.

3. Diagram Rinci (*Level Diagram*)

   Diagram rinci adalah diagram yang menguraikan proses apa saja yang ada dalam diagram nol atau diagram level di atasnya.

2.4.1.2 *State Transition Diagram* (STD)

Menurut Whitten, Bentley, dan Dittman (2004), *State Transition Diagram* adalah alat yang digunakan untuk menggambarkan urutan dan variasi screen yang dapat terjadi selama satu sesi pengguna.

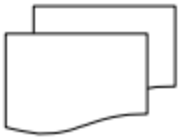
State Transition Diagram juga dapat diartikan sebagai suatu *modeling tools* yang menggambarkan sifat ketergantungan pada waktu dari suatu system. *State Transititon Diagram* terdiri dari:

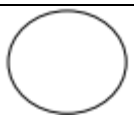
- *State* yaitu keadaan dari sistem, yang dinyatakan dengan kotak.
- *Transition* yaitu arah perpindahan atau transisi dari satu *state* ke *state* lainnya, yang dinyatakan dengan anak panah.
- Kondisi dinyatakan dengan tulisan yang diberi garis bawah.
- Aksi dinyatakan dengan tulisan tanpa garis bawah, biasanya dituliskan di bawah kondisi.

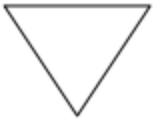
2.4.1.3 *Flowchart*

Menurut Mulyadi (2001), *flowchart* adalah standar yang digunakan analisis sistem untuk menggambarkan suatu sistem

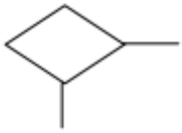
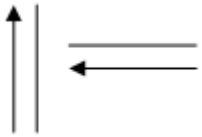
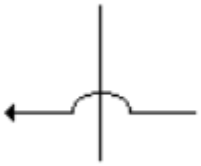
tertentu. Berikut ini adalah simbol-simbol *flowchart* dan kegunaannya masing-masing :

	**Dokumen**	Simbol ini digunakan untuk membuat semua jenis dokumen atau formulir yang digunakan untuk merekam data. Nama dokumen dicantumkan di tengah simbol.
	**Dokumen dan tembusannya**	Simbol ini digunakan untuk menggambarkan dokumen asli dan tembusannya. Nomor lembar dokumen dicantumkan pada sudut kanan atas.
	**Berbagai dokumen**	Simbol ini digunakan untuk menggabungkan berbagai jenis dokumen secara bersamaan di dalam suatu paket. Nama dokumen dituliskan di dalam masing-masing

		simbol dan dokumen dicantumkan di sudut kanan atas simbol.
	**Catatan**	Simbol ini digunakan untuk menggambarkan catatan yang digunakan untuk mencatat data. Nama catatan dicantumkan di dalam simbol ini.
	**Penghubung pada halaman yang sama**	Simbol ini digunakan untuk mengambarkan bagan alir, bagan yang mengalir dari atas ke bawah dan dari kiri ke kanan. Simbol ini digunakan sebagai penghubung pada halaman yang sama.
	**Penghubung pada halaman yang berbeda**	Simbol ini digunakan untuk menguhubungkan bagan alir yang lebih dari satu halaman. Simbol ini menunjukan kemana dan

		bagaimana bagan alir yang tercantum pada halaman berikutnya.
	**Kegiatan manual**	Simbol ini digunakan untuk menggambarkan kegiatan manual yang terjadi, seperti menerima pesanan dan mengisi formulir.
	**Keterangan komentar**	Simbol ini digunakan untuk memberikan keterangan guna memperjelas pesan yang disampaikan dalam aliran data.
	**Arsip sementara**	Simbol ini digunakan untuk menunjukan tempat penyimpanan dokumen sementara. Untuk menunjukkan urutan pengarsipan dokumen, digunakanlah simbol berikut ini:

		A = menurut abjad, N = menurut nomor urut, T = menurut tanggal, kronologis
	**Arsip Permanen**	Simbol ini digunakan untuk menunjukkan tempat penyimpanan dokumen secara permanen.
	***On-line computer procces***	Simbol ini menggambarkan pengolahan data dengan komputer secara *on-line*.
	***Keying (typing, verifying)***	Simbol ini menggambarkan pemasukkan data ke dalam komputer secara *on-line terminal.*
	**Pita magnetik**	Simbol ini menggambarkan arsip komputer yang berbentuk pita magnetik.
	***On-line storage***	Simbol ini menggambarkan arsip

		komputer berbentuk *on-line* (di dalam memori komputer).
	**Keputusan**	Simbol ini menggambarkan keputusan yang harus dibuat dalam proses pengolahan data.
	**Garis alir (*flowline)***	Simbol ini menggambarkan arah garis alir atau proses pengolahan data.
	**Persimpangan garis alir**	Simbol ini digunakan apabila dua buah garis bersimpangan. Apabila dua buah garis bersimpangan maka salah satu garis akan digambarkan melengkung.
	**Pertemuan garis alir**	Simbol ini digunakan apabila dua buah garis bertemu dan salah satu garis mengikuti arus garis lainnya.

	**Mulai/ berakhir**	Simbol ini digunakan untuk menggambarkan dimulai dan berakhirnya suatu sistem.

2.4.2 *Software*

Adapun *software* yang mendukung untuk pembuatan aplikasi ini adalah sebagai berikut:

2.4.2.1 Visual Basic 2010

Menurut Deitel (2011) Visual Basic merupakan pengembangan dari BASIC (*Beginner's All-purpose Symbolic Instruction Code*), yang dikembangkan pada pertengahan tahun 1960 di *Dartmouth College* sebagai bahasa yang digunakan pemula untuk mempelajari teknik-teknik dasar *programmming*. Sebelum Visual Basic dikembangkan, pembuatan aplikasi berbasis Windows merupakan hal yang sangat sulit. Program pada Visual Basic dikembangkan dengan menggunakan sekumpulan *software tools* yang dikenal dengan *Integrated Development Environment* (IDE). Dengan menggunakan Visual Studio 2010 IDE, *programmer* dapat melakukan *write*, *run*, *test*, dan *debug* program dengan cepat dan mudah.

Pengertian Visual Basic secara umum adalah sebuah bahasa pemrograman yang berbasis OOP (*Object Oriented Programming*)

yang digunakan dalam pembuatan aplikasi Windows yang berbasis GUI (*Graphical User Interface*). Namun ada beberapa keterbatasan yang dimiliki oleh visual basic, seperti *inheritance* tidak dapat dilakukan pada *class module* dan *polymorphism* secara terbatas bisa dilakukan dengan deklarasi *class module* yang mempunyai *interface* tertentu. Di lain sisi, Visual Basic tidak *case sensitif* seperti bahasa pemograman lainnya. (diperoleh dari http://id.wikipedia.org/wiki/Visual_Basic, diakses pada 18 Oktober 2012).

2.4.2.2 MySQL

Menurut Welling dan Thomson (2001), MySQL adalah sistem manajemen basis data yang memungkinkan untuk menyimpan, mencari, mengurutkan, dan menerima data dengan cepat. MySQL bersifat *multi user, multi thread server,* dan merupakan bahasa pemrograman yang terstruktur.

Berikut ini adalah keuntungan-keuntungan dari MySQL menurut Welling dan Thomson (2001) yaitu :

- Performa tinggi
- Biaya yang rendah
- Mudah dikonfigurasi dan dipelajari
- Dapat digunakan hampir di semua sistem operasi, seperti Linux dan Windows
- Ketersediaan *source code*

2.4.2.3 phpMyAdmin

phpMyAdmin adalah salah satu aplikasi yang paling banyak digunakan di era *open source*, yang ditulis dalam PHP. phpMyAdmin mendukung berbagai operasi dengan MySQL. Saat ini, hal itu dapat membuat, mengubah dan menghapus database. Menghapus, mengedit, menambahkan kolom, mengeksekusi pernyataan SQL, mengelola *keys* pada bidang, mengelola hak istimewa, mengekspor data ke berbagai format adalah fitur-fitur yang disediakan oleh phpMyAdmin. phpMyAdmin adalah *web-based front-end* untuk mengelola database MySQL dan telah diadopsi oleh sejumlah *Open-Source* distributor. (diperoleh dari http://www.wowebook.be, diakses pada 17 Januari 2013).

2.4.3 Intranet

Menurut Tung (1997) intranet adalah sebuah jaringan dalam perusahaan yang menggunakan protokol-protokol standar internet (TCP/IP), yang digunakan untuk memenuhi kebutuhan perusahaan. Intranet (*internal network*) mulai digunakan pada pertengahan tahun 1995 oleh beberapa penjual produk jaringan yang mengacu pada kebutuhan dalam bentuk web atau aplikasi pada perusahaan.

Keuntungan menggunakan intranet berbeda-beda untuk setiap perusahaan, tetapi terdapat beberapa keuntungan dasar yang sama, yaitu *Wide Area Network* (WAN) dengan jaringan publik. Intranet dan internet menggunakan protocol yang sama, yaitu TCP/IP. Sehingga memungkinkan

penyambungan intranet dengan internet. Dengan demikian suatu perusahaan dapat membuat jaringan yang luas dengan kantor-kantor cabang di kota lain, bahkan di negara lain dengan mudah dan cepat. Dukungan multimedia juga merupakan salah satu keunggulan intranet dan internet, hal ini dikarenakan informasi dialirkan melalui jaringan (*ethernet*) yang memungkinkan data dapat dialirkan dengan cukup cepat.