



Hak cipta dan penggunaan kembali:

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

Copyright and reuse:

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

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Sistem merupakan sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur (O'Brien, 2005). Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Tata Sutabri, 2004).

Informasi merupakan data yang telah diproses atau data yang memiliki arti (McLeod dan Shell, 2004). Menurut Tata Sutabri (2004), informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (Tata Sutabri, 2004).

Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai. Menurut Tata Sutabri (2004), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Dari teori-teori diatas penulis menyimpulkan bahwa sistem informasi merupakan suatu sistem yang berfungsi untuk mengolah data yang dapat menghasilkan informasi yang cepat, tepat dan akurat bagi penggunanya.

2.2 *System Development Life Cycle (SDLC)*

Definisi *System Development Life cycle (SDLC)* menurut (O'brien 2005) adalah aplikasi penerapan dari penemuan permasalahan (*problem solving*) yang didapat dari pendekatan sistem (*system approach*) menjadi pengembangan dari solusi sistem informasi terhadap masalah bisnis. Terdapat tujuh tingkat dari SDLC, yaitu :

1. *Planning*

Pada tahap ini, merupakan tahap perencanaan mengenai definisi sistem, tujuan pengembangan sistem, misi-misi yang dicapai, serta hal-hal yang berhubungan dengan ruang lingkup sistem yang dibangun agar user dapat mengerti makna dari sistem yang akan dikembangkan.

2. Sistem analisis

Sebuah penelitian dari masalah bisnis yang dimana perusahaan berencana untuk menyelesaikannya dengan sebuah sistem informasi.

3. Sistem desain

Desain sebuah sistem yang telah dianalisa dari survei lapangan dan wawancara. Sistem desain terbagi menjadi dua aspek utama untuk sistem baru, yaitu *Logical System Design* dan *Physical System Design*. *Logical System Design* yang menyatakan apapun yang sistem kerjakan, selalu menggunakan spesifikasi *abstract*. *Physical System Design* yang menyatakan bahwa bagaimana sistem bekerja sesuai fungsinya yakni dengan *actual physical specifications*.

4. *Programming*

Dari desain sistem yang telah dibuat, programmer diharapkan dapat membuat program yang sesuai dengan desain sistem menggunakan bahasa-bahasa pengkodean seperti C++, Visual Basic, dan sebagainya.

5. *Testing*

Melakukan pengujian terhadap program yang dibuat untuk kemudian diimplementasi ke dalam bisnis apabila sistem yang dibuat telah cocok sesuai keinginan user.

6. Implementasi

Implementasi adalah sebuah proses dalam mengubah dari sebuah sistem lama menjadi sistem yang baru.

7. *Maintenance/* perbaikan

Tipe- tipe dari *maintenance*, yaitu *debugging* dan *updating*

2.3 Sistem Informasi Pembelian, Penjualan dan Persediaan

2.3.1 Sistem Informasi Pembelian

Sistem akuntansi pembelian digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan. Transaksi pembelian dapat digolongkan menjadi dua: pembelian lokal dan impor. Pembelian lokal adalah pembelian dari pemasok dalam negeri, sedangkan impor adalah pembelian dari pemasok luar negeri (Mulyadi, 2001).

Transaksi pembelian secara umum dapat dibagi menjadi dua, yaitu pembelian tunai dan pembelian kredit.

1. Pembelian Tunai

Pembelian yang pelunasannya dilaksanakan pada saat terjadinya transaksi jual beli.

2. Pembelian Kredit

Pembelian yang proses pelunasannya dilakukan secara berkala sesuai dengan kesepakatan pihak penjual dan pembeli.

2.3.2 Sistem Informasi Penjualan

Menurut Mulyadi (2001), kegiatan penjualan terdiri dari transaksi penjualan barang atau jasa yang bisa dikategorikan menjadi, penjualan tunai, penjualan kredit, penjualan konsinyasi, penjualan *leasing*.

1. Penjualan Tunai

Dalam transaksi penjualan tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli.

Menurut Mulyadi (2001), informasi yang diperlukan oleh manajemen dari transaksi penjualan adalah sebagai berikut:

- a. Jumlah pendapatan penjualan menurut jenis produk atau kelompok produk selama jangka waktu tertentu
- b. Jumlah harga pokok produk yang dijual selama jangka waktu tertentu.
- c. Nama dan alamat pembeli.
- d. Kuantitas produk yang dijual.
- e. Nama wiraniaga yang melakukan penjualan.
- f. Otoritas pejabat yang berwenang.

2.3.3 Sistem Informasi Persediaan

Sistem akuntansi 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 (Mulyadi, 2001).

Sistem dan prosedur yang bersangkutan dengan sistem akuntansi persediaan adalah:

1. Prosedur pencatatan produk jadi.
2. Prosedur pencatatan harga pokok produk jadi yang dijual.
3. Prosedur pencatatan harga pokok produk jadi yang diterima kembali dari pembeli.
4. Prosedur pencatatan tambahan dan penyesuaian kembali harga pokok persediaan produk dalam proses.
5. Prosedur pencatatan harga pokok persediaan yang dibeli.
6. Prosedur pencatatan harga pokok persediaan yang dikembalikan kepada pemasok.

7. Prosedur permintaan dan pengeluaran barang gudang.
8. Prosedur pencatatan tambahan harga pokok persediaan karena pengembalian barang gudang.
9. Sistem penghitungan fisik persediaan.

(Mulyadi, 2001)

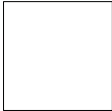

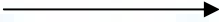
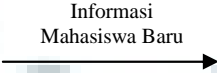
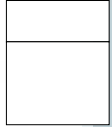
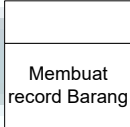

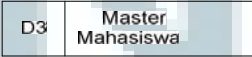
2.4 Alat Perancangan

2.4.1 Data Flow Diagram (DFD) atau Diagram Aliran Data(DAD)

Melalui suatu teknik analisa data terstruktur yang disebut Diagram Aliran Data (DAD), penganalisis sistem dapat merepresentasi proses-proses data di dalam organisasi. Pendekatan aliran data menekankan logika yang mendasari sistem. Dengan menggunakan kombinasi dari empat simbol, penganalisis sistem dapat menciptakan suatu gambaran proses-proses yang bisa menampilkan dokumentasi sistem yang solid (Kenneth E Kendall dan Julie E Kendall, 2003).

Simbol-simbol yang digunakan dalam diagram aliran data dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol dalam diagram aliran data

Simbol	Arti	Contoh
	Entitas	
	Aliran data	
	Proses	
	Penyimpanan data	

Sumber : Kenneth E Kendall dan Julie E Kendall (2003)

1. Menciptakan diagram konteks

Diagram Konteks adalah tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara

keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran data menuju dan dari sistem diketahui penganalisis dari wawancara dengan pengguna dan sebagai hasil analisis dokumen (Kenneth E Kendall dan Julie E Kendall, 2003).

2. Menggambar Diagram 0 (Level Berikutnya)

Diagram 0 adalah pengembangan diagram konteks dan bisa mencakup sampai sembilan proses. Memasukkan lebih banyak proses pada level ini akan terjadi dalam suatu diagram yang kacau yang sulit dipahami. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data-penyimpanan data utama dari sistem (mewakili file-file master) dan semua entitas eksternal dimasukkan ke dalam Diagram 0. dokumen (Kenneth E Kendall dan Julie E Kendall, 2003).

3. Menciptakan Diagram Anak (Tingkat yang Lebih Mendetail)

Setiap proses dalam Diagram 0 bisa dikembangkan untuk menciptakan diagram anak yang lebih mendetail. Proses pada Diagram 0 yang dikembangkan itu disebut *parent process* (proses induk) dan diagram yang dihasilkan disebut *child diagram* (diagram anak). Aturan utama untuk menciptakan diagram anak, keseimbangan vertikal, menyatakan bahwa suatu diagram anak tidak bisa menghasilkan keluaran atau menerima masukan dimana proses induknya juga tidak menghasilkan atau menerima. Semua aliran data yang menuju atau keluar dari proses induk harus ditunjukkan mengalir ke dalam atau keluar dari diagram anak (Kenneth E Kendall dan Julie E Kendall, 2003).

Diagram anak ditetapkan nomor yang sama seperti proses induknya di dalam Diagram 0. Sebagai contoh, proses 3 akan berkembang ke Diagram

3. Proses-proses pada diagram anak diberi nomor dengan menggunakan nomor proses induk, poin desimal, serta sebuah nomor unik untuk setiap proses anak. Pada diagram 3, proses-proses tersebut akan diberi nomor 3.1, 3.2, 3.3, dan seterusnya. Ketentuan ini memungkinkan penganalisis mengikuti rangkaian proses di setiap tingkat pengembangan. Bila Diagram 0 menggambarkan proses-proses 1, 2, dan 3, diagram anak 1, 2, dan 3 semuanya berada pada level yang sama (Kenneth E Kendall dan Julie E Kendall, 2003).

2.4.2 Entity Relationship Diagram

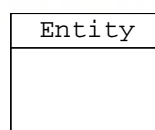
Menurut Dhanta (2009:189), *Entity Relationship Diagram (ERD)* yaitu model konseptual yang menjabarkan hubungan antar penyimpanan data dan hubungan data. Pada *Entity Relationship Diagram (ERD)* terdapat simbol-simbol dengan himpunan relasi yang masing-masing memiliki atribut untuk menjelaskan suatu relasi secara keseluruhan. ERD ini menggambarkan relasi atau hubungan antar entitas yang ada, dimana terdapat 2 jenis hubungan, yaitu:

- a. *Obligatory*: bila semua anggota dari suatu entity harus berpartisipasi atau memiliki hubungan dengan entity yang lain.
- b. *Non-obligatory*: bila tidak semua anggota dari suatu entity harus berpartisipasi atau memiliki hubungan dengan entity yang lain.

Dalam menggambar ERD, ada beberapa komponen yang perlu diperhatikan, yaitu:

a. *Entity*

Entity didefinisikan sebagai sesuatu yang mudah diidentifikasi. Sebuah entity dapat berupa obyek, tempat, orang, konsep atau aktivitas. Pada teknik penggambaran, entity digambarkan dengan kotak segiempat. Setiap kotak diberi label berupa kata benda. Simbol entity dapat dilihat pada Gambar 2.1.



Gambar 2.1. *Entity*

b. Atribut

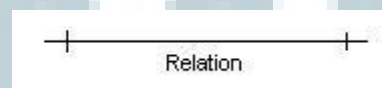
Identifikasi dan deskripsi dari entity dijelaskan oleh atribut-atributnya (karakteristik entity). Sebuah atribut didefinisikan sebagai penjelasan-penjelasan dari entity yang membedakannya dengan entity yang lain. Selain itu, atribut juga merupakan sifat-sifat dari sebuah entity. Sebagai contoh, entity Proficiency mempunyai atribut IDProficiency, Name, dan atribut lainnya. Contoh atribut dapat dilihat pada Gambar 2.2.

Proficiency
<u>IDProficiency</u>
Name
Date
Time

Gambar 2.2. Atribut

c. Relasi

Relasi adalah penghubung antara suatu entity dengan entity yang lain dan merupakan bagian yang sangat penting dalam mendesain database. Simbol relasi dapat dilihat pada Gambar 2.3.

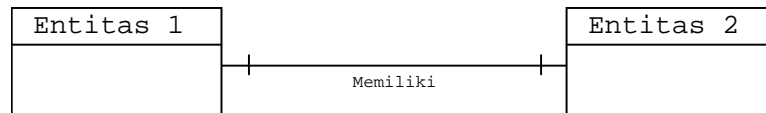


Gambar 2.3 Relasi

Ada tiga macam relasi, antara lain:

a. *One-to-One*

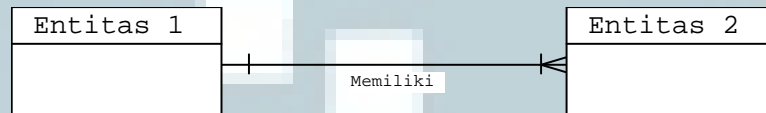
Pada bentuk relasi *One-to-One*, satu anggota entity memiliki hubungan dengan satu anggota entity pada kelas yang berbeda. Simbol relasi *One-to-One* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Relasi *One to One*

b. *One-to-Many*

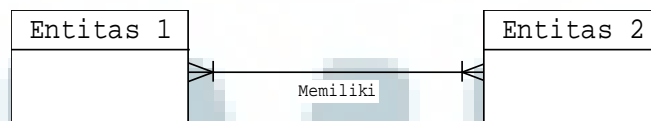
Pada bentuk relasi *One-to-Many*, satu anggota entity bisa memiliki hubungan dengan beberapa anggota entity pada kelas yang berbeda. Sama halnya dengan *One-to-One*, pada relasi yang satu ini juga terbagi ke dalam 2 jenis hubungan, yaitu: obligatory dan non-obligatory. Simbol relasi *One-to-Many* dapat dilihat pada Gambar 2.5.



Gambar 2.5. Relasi *One to Many*

c. *Many-to-Many*

Pada bentuk relasi *Many-to-Many*, beberapa anggota entity dapat memiliki hubungan dengan beberapa anggota entity lainnya. Dalam relasi ini juga terdapat dua jenis hubungan, yaitu: obligatory dan non-obligatory. Simbol relasi *Many-to-Many* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Relasi *Many to Many*

Sebagai contoh, hubungan antara dosen dengan mata kuliah yang diajarkannya. Maka bentuk diagram adalah sebagai berikut :



Gambar 2.7 Contoh ERD

Pada gambar diatas, Dosen dan Mata Kuliah merupakan entitas, sedangkan Mengajar merupakan relasi antara Dosen dan Mata Kuliah. Dari contoh ini dapat terjadi beberapa relasi dengan kondisi-kondisi tertentu sebagai berikut :


1. Seorang dosen hanya dapat mengajar satu mata kuliah dan satu mata kuliah hanya dapat diajar oleh seorang dosen saja. Relasi dengan kondisi seperti ini dinamakan relasi *One-to-One* (1 : 1).
2. Seorang dosen dapat mengajar lebih dari satu mata kuliah dan satu mata kuliah hanya dapat diajar oleh seorang dosen saja. Relasi dengan kondisi seperti ini dinamakan relasi *One-to-Many* (1 : n).
3. Seorang dosen hanya dapat mengajar satu mata kuliah dan satu mata kuliah dapat diajar lebih dari satu orang dosen. Relasi dengan kondisi seperti ini dinamakan relasi *Many-to-One* (n : 1).
4. Seorang dosen dapat mengajar lebih dari satu mata kuliah dan satu mata kuliah dapat diajar lebih dari satu orang dosen saja. Relasi dengan kondisi seperti ini dinamakan relasi *Many-to-Many* (m : n).

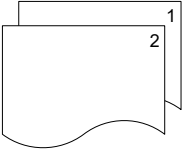
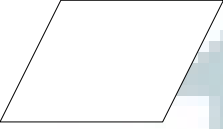




2.4.3 Flow of Document

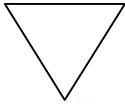





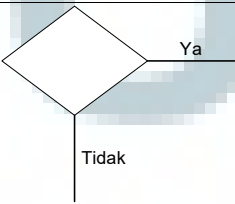
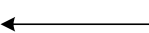
Sistem akuntansi dapat dijelaskan dengan menggunakan bagan alir dokumen. Tabel 2.2 melukiskan simbol-simbol standar yang digunakan oleh analis sistem untuk membuat bagan alir dokumen yang menggambarkan sistem tertentu (Mulyadi, 2001).


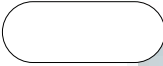
Berikut ini adalah simbol-simbol standar dengan maknanya masing-masing:

Tabel 2.2 Simbol-simbol dalam bagan alir dokumen

Simbol	Arti	Keterangan
	Dokumen	Simbol ini menggambarkan semua jenis dokumen, yang merupakan formulir yang digunakan untuk merekam data terjadinya suatu transaksi

	<p>Dokumen dan tembusannya</p>	<p>Simbol ini menggambarkan dokumen asli dan tembusannya. Nomor lembar dokumen dicantumkan di sudut kanan atas.</p>
	<p>Catatan</p>	<p>Simbol ini menggambarkan catatan akuntansi yang digunakan untuk mencatat data yang direkam sebelumnya di dalam dokumen atau formulir.</p>
	<p>Penghubung pada halaman yang sama</p>	<p>Karena keterbatasan ruang halaman kertas untuk menggambar, diperlukan simbol penghubung untuk memungkinkan aliran dokumen berhenti di suatu lokasi pada halaman tertentu dan kembali berjalan di lokasi lain pada halaman yang sama.</p>
	<p>Penghubung pada halaman yang berbeda</p>	<p>Simbol penghubung ini menunjukkan bagaimana bagan alir yang tercantum pada halaman tertentu terkait dengan bagan alir yang tercantum pada halaman yang lain.</p>
	<p>Kegiatan manual</p>	<p>Simbol ini menggambarkan kegiatan manual, seperti: menerima order dari pembeli, mengisi formulir.</p>
	<p>Keterangan, komentar</p>	<p>Sistem ini memungkinkan ahli sistem menambahkan keterangan untuk memperjelas pesan yang disampaikan dalam bagan alir.</p>

Simbol	Arti	Keterangan
	Arsip sementara	Simbol ini menunjukkan tempat penyimpanan dokumen, seperti almari arsip dan kotak arsip. Untuk menunjukkan urutan pengarsipan dokumen digunakan simbol berikut ini: A = menurut abjad N = menurut nomor urut T = kronologis, menurut tanggal
	Arsip permanen	Simbol ini menggambarkan arsip permanen yang merupakan tempat penyimpanan dokumen yang tidak akan diproses lagi.
	<i>On-line computer process</i>	Simbol ini menggambarkan pengolahan data dengan komputer secara <i>on-line</i> .
	<i>Keying (typing, verifying)</i>	Simbol ini menggambarkan pemasukan data ke dalam komputer melalui <i>on-line terminal</i> .
	Pita magnetik (<i>magnetic tape</i>)	Simbol ini menggambarkan arsip komputer yang berbentuk pita magnetik
	<i>On-line storage</i>	Simbol ini menggambarkan arsip komputer yang berbentuk <i>on-line</i> (di dalam <i>memory</i> komputer)
	Keputusan	Simbol ini menggambarkan keputusan yang harus dibuat dalam proses pengolahan data.
	Garis alir (<i>flowline</i>)	Simbol ini menggambarkan arah proses pengolahan data.

	<p>Persimpangan garis alir</p>	<p>Jika dua garis alir bersimpangan, untuk menunjukkan arah masing-masing garis, salah satu garis dibuat sedikit melengkung tepat pada persimpangan kedua garis tersebut.</p>
	<p>Mulai/berakhir (<i>terminal</i>)</p>	<p>Simbol ini untuk menggambarkan awal dan akhir suatu sistem akuntansi.</p>

Sumber: Mulyadi (2001)

2.4.4 *Data Dictionary* atau Kamus Data

Kamus data adalah suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Kamus data merupakan hasil referensi data mengenai data, suatu data yang disusun oleh penganalisis sistem untuk membimbing mereka selama melakukan analisis dan desain. Sebagai suatu dokumen, kamus data mengumpulkan dan mengkoordinasikan istilah-istilah data tertentu, dan menjelaskan apa arti setiap istilah yang ada (Kenneth E Kendall dan Julie E Kendall, 2003).

Simbol-simbol yang digunakan dalam kamus data dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol kamus data

Notasi	Keterangan
=	Terdiri dari
+	Dan
{ }	Elemen-elemen repetitif (kelompok berulang)
[]	Salah satu dari dua situasi tertentu
()	Pilihan (boleh dikosongkan)

Sumber : Kenneth E Kendall dan Julie E Kendall (2003)

2.5.5 *Data Base* atau Basis Data

Basis data tidak hanya merupakan kumpulan file. Lebih dari itu, basis data adalah pusat sumber data yang caranya dipakai oleh banyak pemakai

untuk berbagai aplikasi. Inti dari basis data adalah *database management system* (DBMS), yang membolehkan pembuatan, modifikasi, dan pembaharuan basis data; mendapatkan kembali data; dan membangkitkan laporan (Kenneth E Kendall dan Julie E Kendall, 2003).

Menurut Kenneth E Kendall dan Julie E Kendall (2003), tujuan basis data yang efektif yaitu:

1. Memastikan bahwa data dapat dipakai di antara pemakai untuk berbagai aplikasi.
2. Memelihara data baik keakuratan maupun kekonsistennya.
3. Memastikan bahwa semua data yang diperlukan untuk aplikasi sekarang dan yang akan datang akan disediakan dengan cepat.
4. Membolehkan basis data untuk berkembang dan kebutuhan pemakai untuk berkembang.
5. Membolehkan pemakai untuk membangun pandangan personalnya tentang data tanpa memperhatikan cara data disimpan secara fisik.

2.5 Normalisasi

Normalisasi adalah transformasi tinjauan pemakai yang kompleks dan data tersimpan ke sekumpulan bagian-bagian struktur data yang kecil dan stabil. Di samping menjadi lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah diatur daripada struktur data lainnya. (Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 145)

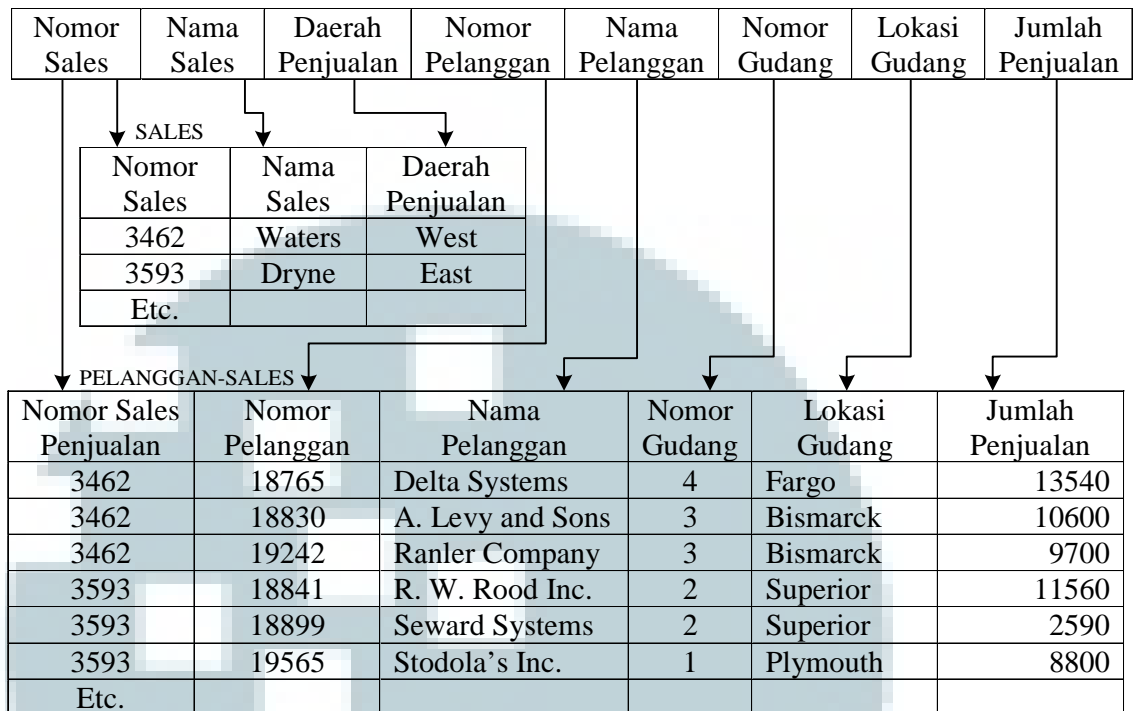
Tahapan normalisasi yaitu:

1. Tahapan Pertama

Tahap pertama dari proses meliputi menghilangkan semua kelompok terulang dan mengidentifikasi kunci utama. Untuk mengerjakannya, hubungan perlu dipecah ke dalam dua atau lebih hubungan. Pada titik ini, hubungan mungkin sudah menjadi bentuk normalisasi ketiga, bahkan lebih banyak tahap akan diperlukan untuk mentransformasi hubungan ke bentuk normalisasi ketiga. (Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 145)

Contoh:

LAPORAN-PENJUALAN



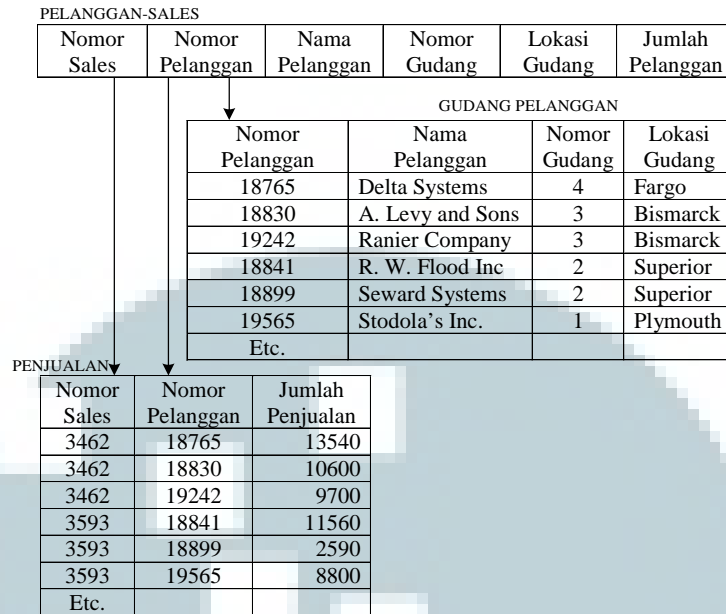
Gambar 2.8 Hubungan tidak normal yang asli dari LAPORAN-PENJUALAN dipisah ke dalam dua hubungan, SALES (3NF) dan PELANGGAN-SALES (1NF)

Sumber : Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 150

2. Tahapan Kedua

Tahap kedua menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain. (Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 145)

Contoh :



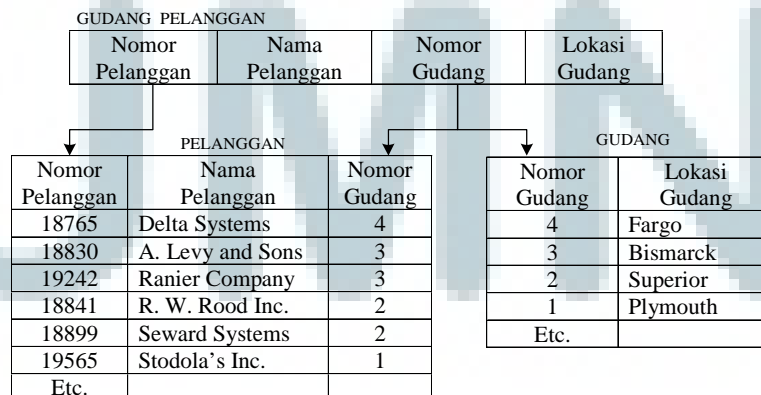
Gambar 2.9 Hubungan PELANGGAN-SALES dipisah ke dalam hubungan yang dinamakan GUDANG PELANGGAN (2NF) dan hubungan yang dinamakan PENJUALAN (1NF)

Sumber : Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 152

3. Tahapan Ketiga

Tahap ketiga mengubah ketergantungan transitif manapun. Suatu ketergantungan transitif adalah sesuatu di mana atribut bukan kunci tergantung pada atribut bukan kunci lainnya. (Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 145)

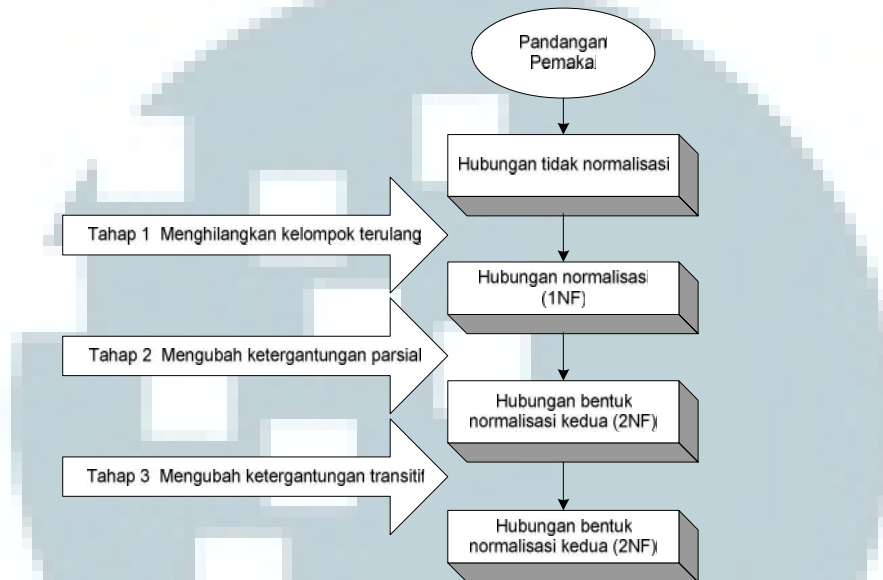
Contoh:



Gambar 2.10 Hubungan GUDANG-PELANGGAN dipisah ke dalam dua hubungan yang dinamakan PELANGGAN (1NF) dan GUDANG (1NF)

Sumber : Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 153

Berikut adalah gambaran tahapan yang dilakukan dalam normalisasi:



Gambar 2.11 Tahapan Normalisasi

Sumber : Kenneth E Kendall & Julie E Kendall, 2003, Jilid 2, 146

2.6 MySql

Menurut (Nugroho, 2004) MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL juga dapat berperan sebagai *client/server*, yang *pen source* dengan kemampuan dapat berjalan baik di OS (*Operating System*) manapun. Selain itu *database* ini memiliki kelebihan dibanding *database* lain, diantaranya adalah:

1. MySQL sebagai *Database Management System (DBS)*
2. MySQL sebagai *Relation Database Management System (RDBMS)*

3. MySQL adalah sebuah *software database* yang bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
4. MySQL merupakan *database server*, jadi dengan menggunakan *database* ini, dapat dihubungkan ke media internet sehingga dapat diakses dari jauh.
5. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query* yang mengakses *database* pada *server*.
6. Mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
7. Mampu menyimpan data yang berkapasitas besar hingga berukuran *gigabyte* sekalipun.
8. Memiliki kecepatan dalam pembuatan tabel maupun *update* tabel.
9. Menggunakan bahasa permintaan standar yang bernama SQL (*Structure Query Language*)

UMMN