

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem merupakan suatu kesatuan yang terdiri dari beberapa unit kesatuan yang saling bekerja dan saling ketergantungan yang akan digunakan untuk mencapai tujuan tertentu [3].

Adapun ciri-ciri sistem, yaitu [3]:

1. Terdiri dari beberapa komponen yang saling berhubungan
2. Memiliki lingkungan luar
3. Memiliki *interface*
4. Terdiri dari masukan, pengolahan, dan keluaran

Setiap sistem memiliki karakteristik yang meliputi [3]:

1. Komponen Sistem (*Components*)

Sistem terdiri dari komponen-komponen yang saling berinteraksi yang membuat suatu kesatuan, bekerja untuk mencapai suatu tujuan yang disebut sebagai subsistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungannya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem merupakan batasan sistem yang berada di luar yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan dan merugikan yang harus dijaga dan dikendalikan, jika tidak akan mengganggu kelangsungan hidup suatu sistem.

4. Penghubung Sistem (*Interface System*)

Penghubung sistem merupakan media yang menghubungkan antara sistem dengan subsistem lain. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. *Output* dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut, sehingga dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input System*)

Masukan sistem adalah energi yang akan dimasukkan ke dalam sistem seperti pemeliharaan an sinyal.

6. Keluaran Sistem (*Output System*)

Keluaran sistem adalah hasil dari masukan sistem yang telah diolah dan menjadi keluaran yang dapat digunakan kembali sebagai sisa pembuangan.

7. Pengolahan Sistem (*Process System*)

Merupakan proses sistem untuk mengolah energi masukan menjadi sisa pembuangan atau keluaran sistem.

8. Sasaran Sistem (*Objective System*)

Sasaran sistem adalah tujuan (*Goal*) yang digunakan untuk menentukan *input* yang dibutuhkan sistem dan keluaran yang dihasilkan. Sistem dapat dikatakan berhasil jika tujuannya telah tercapai.

2.1.2 Informasi

Informasi adalah data masukan yang telah diproses atau diolah sedemikian rupa menjadi suatu bentuk yang memiliki arti dan fungsi yang berperan penting bagi penerima dalam meningkatkan pengetahuan [3].

Kualitas suatu informasi tergantung pada 3 (tiga) hal, yaitu [3]:

1. Akurat (*Accurate*)

Informasi dikatakan berkualitas ketika informasi tersebut bebas dari kesalahan-kesalahan dan tidak menyesatkan serta harus jelas mencerminkan maksudnya. Informasi harus akurat untuk menghindari kemungkinan terjadinya gangguan yang akan merusak atau mengubah informasi tersebut.

2. Tepat pada waktunya (*time lines*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang telah melebihi batas waktu tidak memiliki nilai karena informasi adalah landasan dalam pengambilan keputusan, jika pengambilan keputusan terlambat akan berakibat fatal terhadap organisasi.

3. Relevan (*Relevance*)

Relevan berarti informasi yang diterima atau diberikan harus memiliki manfaat. Relevansi informasi berbeda-beda terhadap setiap orangnya.

2.1.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang digunakan untuk mempertemukan kebutuhan pengolahan transaksi harian yang dapat mendukung fungsi operasi organisasi dalam menyediakan informasi yang diperlukan dalam pengambilan keputusan [3].

Komponen-komponen Sistem informasi meliputi [4]:

1. Orang

Orang atau personil yang berperan dalam penggunaan sistem informasi yang terdiri dari operator komputer, analisis sistem, *programmer*, personil data *entry*, dan manajer sistem informasi.

2. Perangkat Keras(*Hardware*)

Perangkat keras terdiri dari komputer (Pusat pengolah data, unit masukan/keluaran), peralatan penyimpanan data dan terminal masukan/keluaran.

3. Perangkat lunak(*Software*)

Perangkat lunak dibagi menjadi 3(tiga) kategori yaitu:

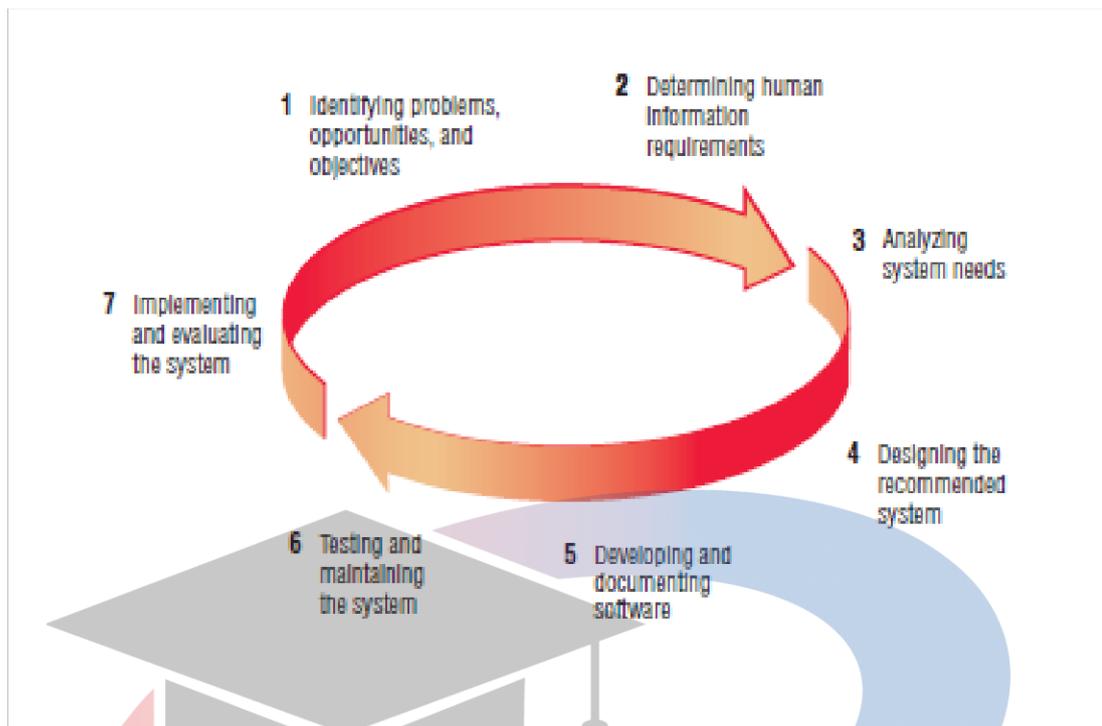
- a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang digunakan untuk menjalankan pengoperasian sistem komputer.
- b. Aplikasi perangkat lunak umum, seperti aplikasi model analisis dan keputusan.
- c. Aplikasi perangkat lunak yang terdiri dari program yang secara spesifik digunakan pada setiap aplikasi.

4. Data

Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi

2.2 System Development Life Cycle

System Development Life Cycle merupakan sebuah pendekatan yang dilakukan secara bertahap yang digunakan untuk menganalisis dan merancang sebuah sistem dengan menggunakan siklus yang spesifik terhadap kegiatan pemakai [5].



Gambar 2.1 Tujuh Tahap *System Development Life Cycle*

Berikut tujuh fase *System Development Life Cycle* menurut Kendall [5]:

1. *Identifying problems, opportunities, and objectives*

Pada fase yang pertama ini, seorang analis harus memperhatikan dengan benar dalam mengidentifikasi masalah, peluang, dan tujuan. Tahap ini merupakan tahap yang sangat penting karena akan menentukan keberhasilan sebuah proyek, karena tidak ada seorang pun yang ingin membuang waktu untuk mengatasi sebuah masalah. Peluang adalah situasi dimana penganalis yakin bahwa peningkatan dapat dilakukan melalui penggunaan komputer yang telah terperinci. Merebut peluang dapat memungkinkan bisnis untuk mendapatkan keunggulan kompetitif atau menetapkan standar-standar industri. Menentukan tujuan juga merupakan komponen yang penting pada tahap ini, dimana analis harus mampu menemukan apa yang terjadi dalam bisnis. Kemudian analis dapat melihat beberapa aspek aplikasi sistem informasi yang dapat membantu bisnis dalam mencapai tujuan.

2. *Determining human information requirements*

Fase berikutnya, analis menentukan kebutuhan informasi bagi pengguna dengan menggunakan bantuan berbagai alat yang mudah dipahami bagaimana seorang pengguna dapat berinteraksi dengan baik dalam konteks pekerjaan dengan sistem informasi yang ada. Analis akan menggunakan metode interaktif seperti, wawancara, kuesioner untuk mengetahui lebih tentang kebutuhan informasi pengguna.

3. *Analyzing system needs*

Perangkat dan teknik khusus membantu penganalis dalam menentukan kebutuhan seperti diagram aliran data (DFD) untuk memetakan *input*, *process*, dan *output* dari diagram aktivitas untuk mengatur sistem dalam bentuk grafik yang terstruktur. Dari aliran data, kamus data dikembangkan yang berisikan semua item data yang akan digunakan dalam sistem. Terdapat tiga metode utama yang digunakan dalam menganalisis struktur keputusan seperti, *structured English*, tabel keputusan (*decision tabel*), dan pohon keputusan (*decision tree*).

4. *Designing the recommended System*

Mendesain sistem yang direkomendasikan. Dalam melakukan desain sistem seorang analis harus menggunakan informasi yang telah dikumpulkan sebelumnya agar dapat menyelesaikan desain logis dari sistem informasi. Analis kemudian merancang prosedur yang dapat digunakan oleh pengguna untuk mengolah data secara akurat menjadi informasi yang benar. Selain itu analis harus menyediakan sebuah *output*, *input*, struktur *file*, program *file*, program, prosedur, perangkat keras, dan perangkat lunak yang diperlukan untuk mendukung sistem informasi.

5. *Developing and documenting software*

Mengembangkan dan mendokumentasikan perangkat lunak. Pada fase ini sistem analisis harus bekerja sama dengan *programmer* untuk mengembangkan perangkat lunak sesuai dengan yang dibutuhkan. Dalam fase ini sistem analis juga harus bekerja dengan pengguna (*user*) dalam membuat dokumentasi perangkat lunak melalui, prosedur manual, *online help*, hingga *file* seperti *Read Me* yang disisipkan pada perangkat lunak yang akan dibuat. Dengan terlibatnya pengguna dalam pembuatan dokumentasi dapat menjawab pertanyaan yang ada sehingga dapat diselesaikan bersama-sama dengan analis. *Programmer* berperan penting dalam menjamin kualitas program dengan menyediakan desain yang menjelaskan kelengkapan sebuah program kepada tim pembangun program.

6. *Testing and Maintaining the system*

Sebelum sistem dapat digunakan, maka harus terlebih dahulu melakukan uji coba. *Testing* bertujuan untuk mengurangi banyaknya masalah yang fatal sebelum sistem dapat digunakan oleh pengguna pada umumnya. Beberapa kesalahan tersebut dapat diselesaikan oleh *Programmer* saja, tidak jarang pula masalah yang ada harus diselesaikan oleh analis dan *programmer*. Rangkaian uji coba pertama dijalankan dengan menggunakan sampel data yang kemudian dibandingkan dengan data sebenarnya dari sistem sebelumnya. Pemeliharaan sistem dan dokumentasinya dimulai pada fase ini dan dilakukan secara rutin sepanjang penggunaan sistem tersebut. Beberapa perawatan sistem yang akan sering terjadi

seperti (*program updates*) dapat dilakukan secara otomatis untuk menghemat biaya dan waktu.

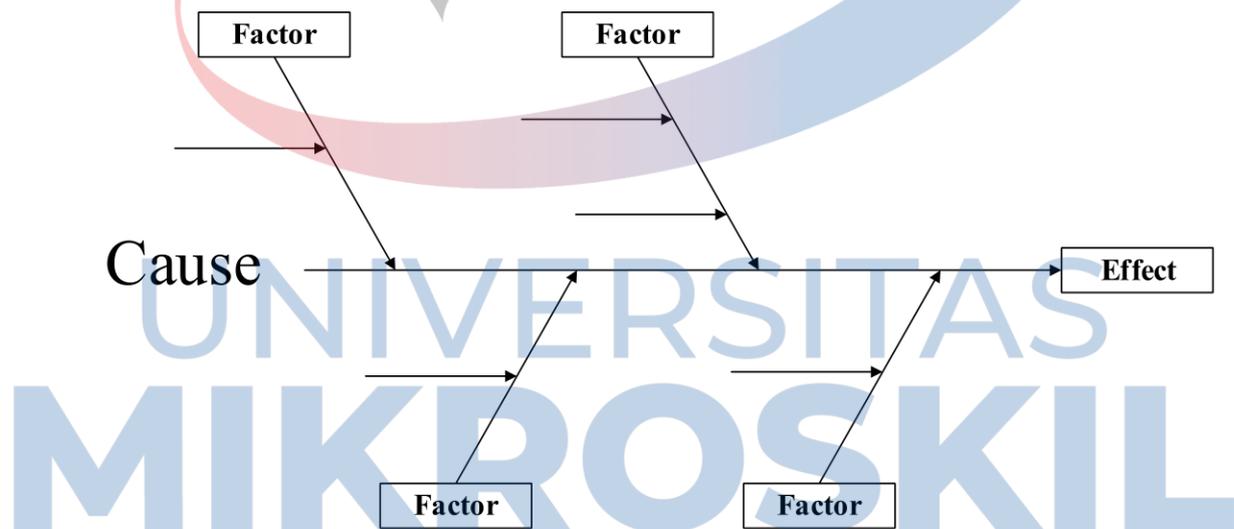
7. *Implementing and evaluating the system*

Fase ini merupakan fase terakhir dalam pengembangan sistem menggunakan metode *System Development Life Cycle* (SDLC). Proses implementasi dilakukan oleh analis dengan melakukan beberapa pelatihan dan merencanakan perubahan yang baik dari sistem yang lama ke sistem yang baru. Evaluasi terdiri atas penilaian sejauh mana suatu sistem yang telah dibangun dapat dijalankan.

2.3 Alat Bantu Analisis dan Perancangan

2.3.1 Fishbone Diagram

Fishbone Diagram merupakan suatu diagram yang menunjukkan suatu hubungan sebab-akibat yang dapat membantu untuk mengidentifikasi penyebab dari suatu masalah yang selanjutnya bisa diambil sebuah tindakan yang tepat. *Fishbone Diagram* dapat digunakan untuk menganalisis faktor-faktor yang menjadi penyebab kegagalan [6].



Gambar 2.2 Struktur *Fishbone Diagram*

Fungsi dari *Fishbone Diagram* (tulang ikan) sebagai berikut [7]:

1. Membantu mengidentifikasi akar penyebab dari suatu masalah
2. Membantu menemukan solusi suatu masalah.
3. Membantu dalam pencarian fakta
4. Mengidentifikasi tindakan untuk mencapai hasil yang diinginkan.
5. Menghasilkan pemikiran yang baru.

Langkah-langkah pembuatan *Fishbone Diagram*, yaitu [7]:

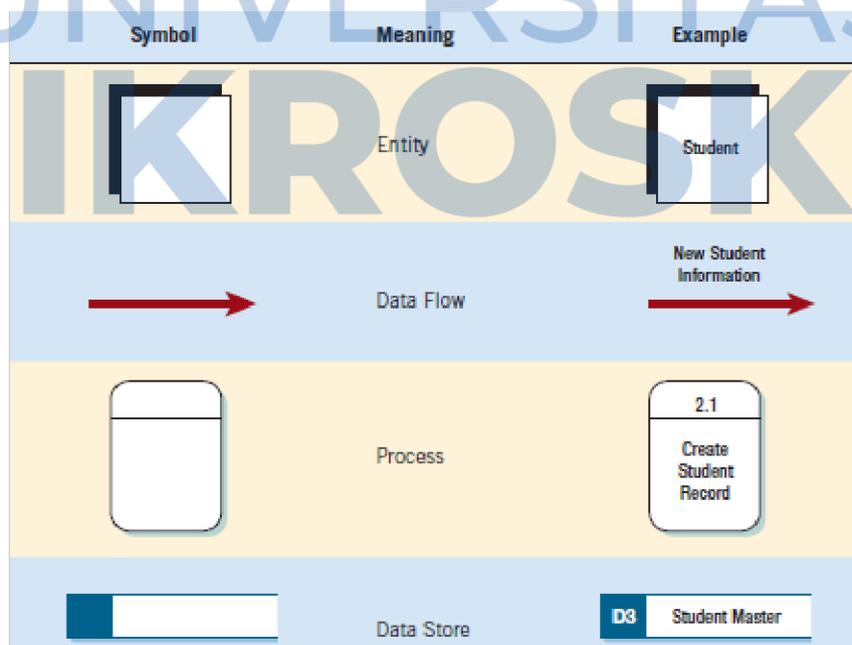
1. Tentukan masalah yang akan diperbaiki, tulis atau gambarkan pada kepala ikan.
2. Tentukan faktor atau penyebab utama yang mempengaruhi masalah.
3. Tentukan penyebab sekunder yang mempengaruhi penyebab utama.
4. Tentukan penyebab utama dari masalah.

2.3.2 Data Flow Diagram (DFD)

Data flow diagram (DFD) adalah model logika data yang digunakan dalam proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut [8].

Berikut adalah beberapa aturan dasar yang harus diikuti dalam menggambar *data flow diagram* [5]:

1. DFD harus memiliki setidaknya satu proses, dan tidak boleh memiliki objek yang berdiri sendiri atau objek yang terhubung dengan dirinya sendiri.
2. Suatu proses harus menerima setidaknya satu aliran data yang masuk ke dalam proses dan membuat setidaknya satu aliran data keluar dari proses.
3. Penyimpanan data harus terhubung ke setidaknya satu proses.
4. Entitas eksternal tidak boleh terhubung satu sama lain. Meskipun berkomunikasi secara mandiri, bahwa komunikasi bukan bagian dari sistem yang kami rancang menggunakan DFD.



Gambar 2.3 Simbol *Data Flow Diagram*

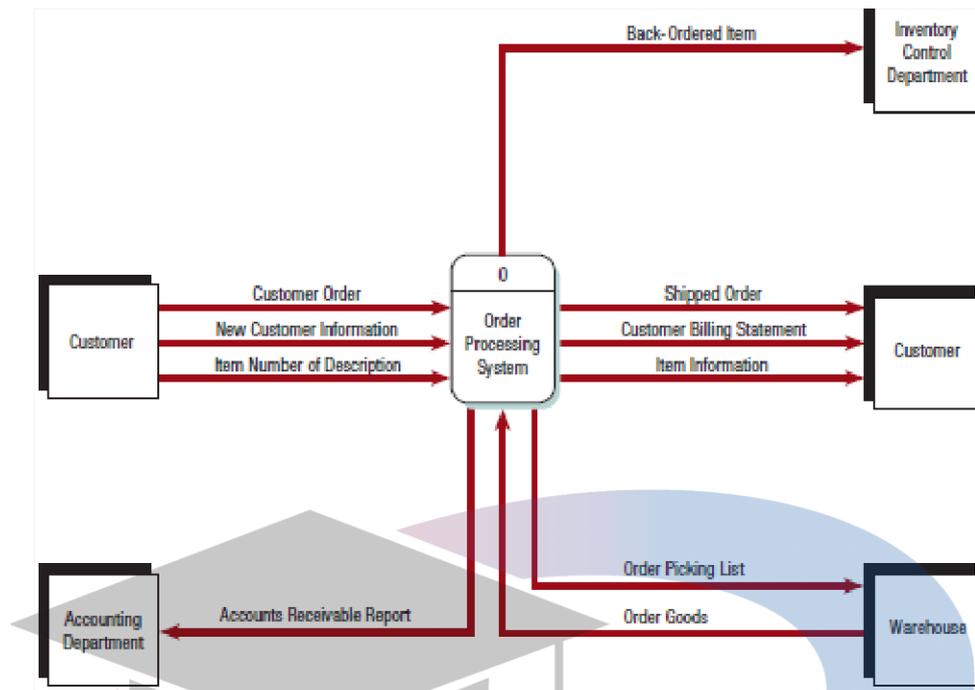
Langkah-langkah pengembangan *data flow diagram*, antara lain [5]:

1. Buatlah daftar kegiatan usaha dan gunakan untuk menentukan berbagai
 - a. Entitas eksternal
 - b. Aliran data
 - c. Proses
 - d. Penyimpanan data
2. Buat diagram konteks yang menunjukkan entitas dan data eksternal mengalir ke dan dari sistem. Jangan tampilkan proses detail apapun atau penyimpanan data.
3. Gambar diagram 0, level selanjutnya. Tunjukkan proses, tetapi pertahankan. Tampilkan penyimpanan data pada level ini.
4. Buat diagram anak untuk setiap proses pada diagram 0.
5. Periksa kesalahan dan pastikan label yang ditetapkan untuk masing-masing proses dan aliran data yang bermakna.
6. Mengembangkan diagram aliran data fisik dari aliran data logis diagram. Bedakan antara proses manual dan otomatis.
7. Partisi diagram aliran data fisik dengan memisahkan atau mengelompokkan bagian dari diagram untuk memfasilitasi pemrograman dan penerapan.

Contoh *Data Flow Diagram*, yaitu [5]:

1. Contoh DFD Konteks

Diagram ini menunjukkan ORDER PROCESSING SYSTEM di tengah, tidak ada proses yang dijelaskan secara rinci dalam diagram tingkat konteks dan lima entitas eksternal (dua entitas terpisah yang keduanya disebut PELANGGAN benar-benar satu dan sama). Aliran data yang datang dan pergi ke entitas eksternal juga ditampilkan (misalnya, CUSTOMER ORDER dan ORDER PICKING LIST).

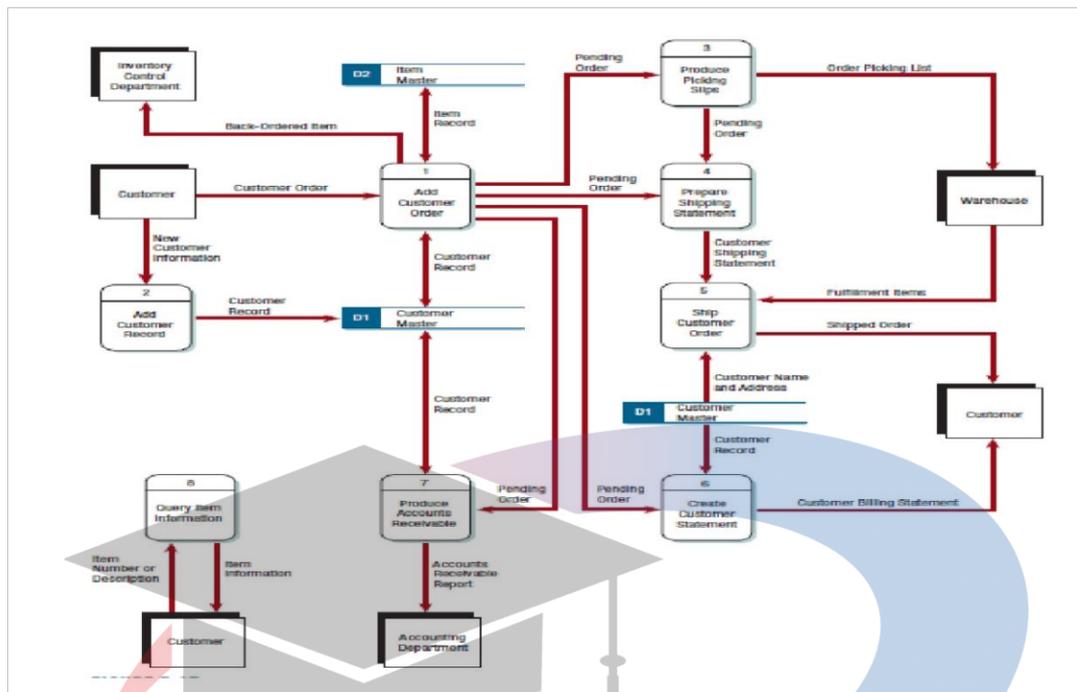


Gambar 2.4 Contoh DFD Konteks

2. Contoh DFD level 0

Diagram ini menunjukkan proses pemesanan dimana diagram 0 bertujuan untuk menjaga proses tetap pada umumnya agar tidak merumitkan pembuatan diagram. Diagram 0 menggambarkan aliran data di antara dan ke entitas eksternal yang sama pada diagram konteks.

UNIVERSITAS MIKROSKIL

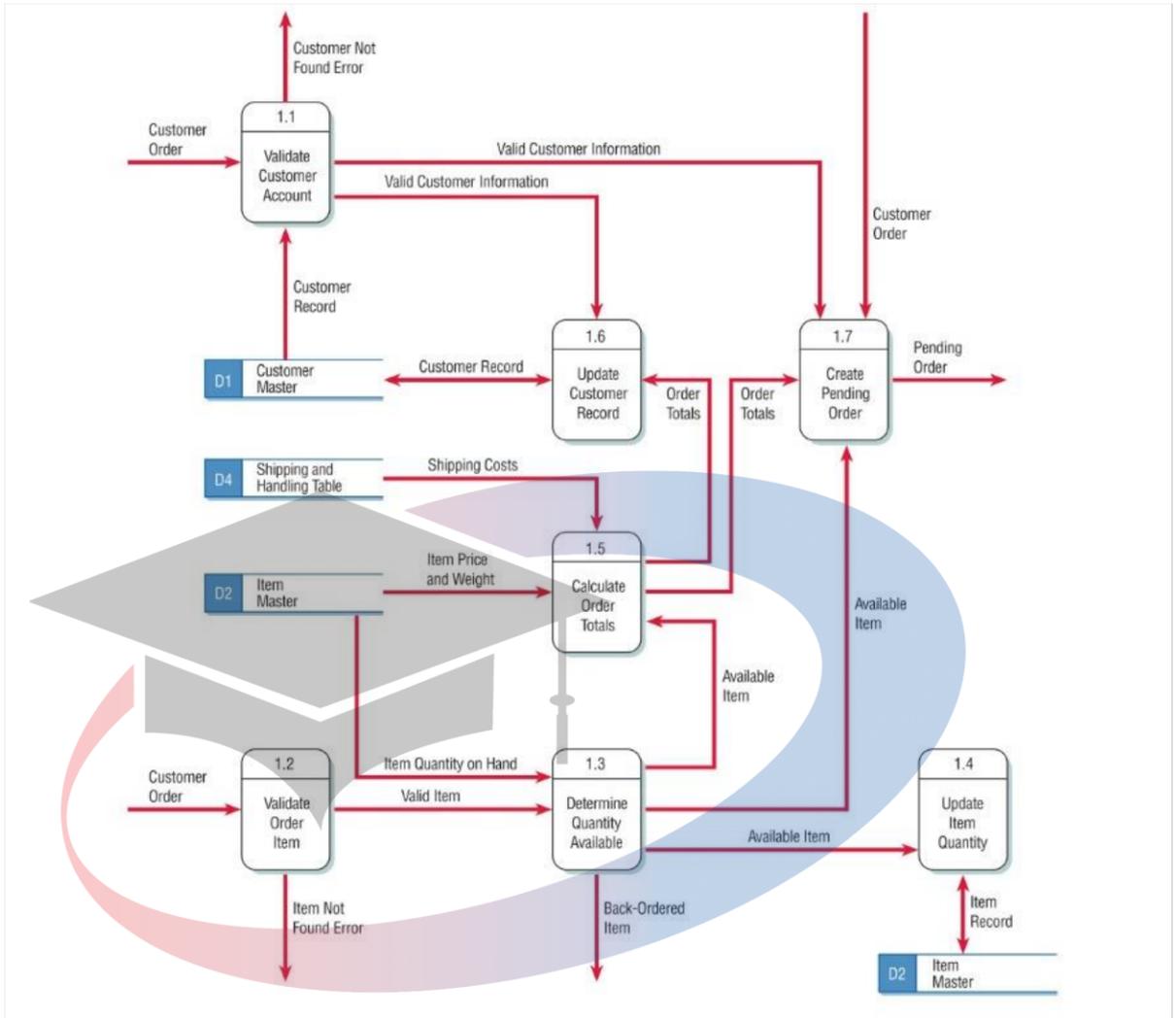


Gambar 2.5 Contoh DFD Level 0

3. Contoh DFD Level 1 Proses 1.0

Proses diagram level 1 menggambarkan logika yang diperlukan untuk menghasilkan keluaran secara lebih rinci. Saat menggambar diagram level 1, buat daftar subproses terlebih dahulu. Sebuah proses seperti ADD CUSTOMER ORDER dapat memiliki subproses sebanyak tujuh. Subproses dihubungkan satu sama lain dan juga ke penyimpanan data bila diperlukan.

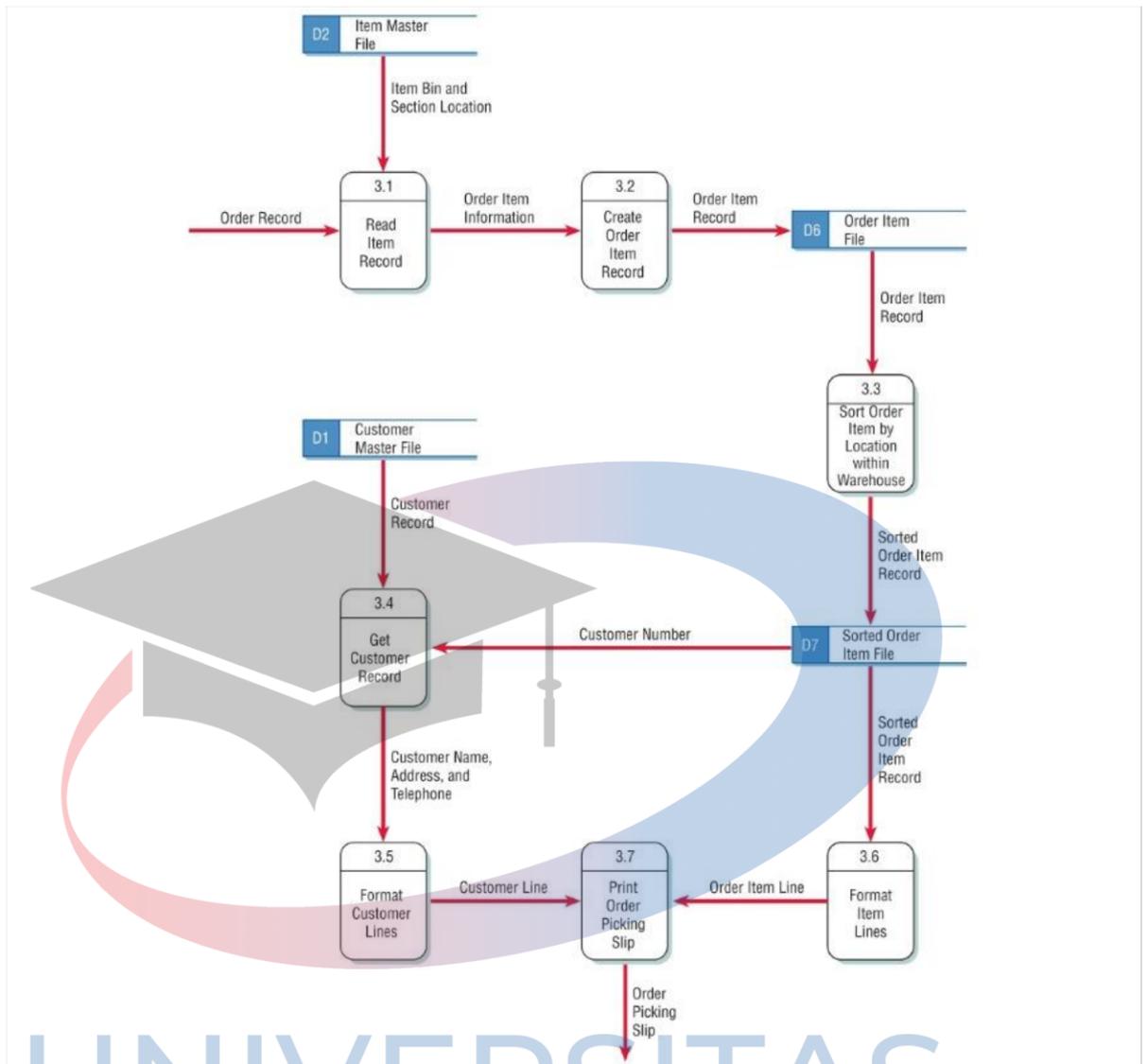
UNIVERSITAS
MIKROSKIL



Gambar 2.6 Contoh DFD Level 1 Proses 1.0

4. Contoh DFD Level 1 Proses 3.0

Gambar 2.7 merupakan contoh DFD level 1 proses 3.0 yaitu PRODUCE PICKING SLIPS yang memungkinkan untuk mengidentifikasi proses memindai kode batang, menampilkan layar, menemukan catatan, dan membuat serta memperbaharui *file*.



Gambar 2.7 Contoh DFD Level 1 Proses 3.0

2.3.3 Kamus Data

Kamus data adalah dokumen yang digunakan untuk mengumpulkan dan mengoordinasikan istilah data tertentu, dan menginformasikan arti setiap istilah bagi orang yang berbeda dalam organisasi. Salah satu alasan penting untuk memelihara kamus data adalah untuk menjaga kebersihan data. Artinya data harus konsisten. Selain membantu menghilangkan redundansi, kamus data dapat digunakan untuk [5]:

1. Validasi diagram aliran data untuk kelengkapan dan akurasi.
2. Memberikan titik awal untuk mengembangkan layar dan laporan.
3. Menentukan isi data yang disimpan dalam *file*.
4. Mengembangkan logika untuk proses diagram aliran data.

Berikut adalah notasi-notasi yang digunakan dalam kamus data, yaitu [5]:

Tabel 2.1 Notasi Kamus Data

Simbol	Keterangan
=	Terdiri dari, mendefinisikan, diuraikan menjadi
+	Dan
()	Opsional
{ }	Perulangan
[]	Situasi baik.

Contoh kamus data, yaitu [5]:

Pesanan pelanggan = Nomor Pelanggan +
 Nama Pelanggan +
 Alamat +
 Telepon +
 Nomor Katalog +
 Tanggal Pemesanan +
 {Item Pesanan yang Tersedia} +
 Jumlah Barang Dagangan +
 (Pajak) +
 Pengiriman dan Penanganan +
 Jumlah Pesanan +
 Cara pembayaran +
 (Jenis Kartu Kredit) +
 (Nomor Kartu Kredit) +
 (Tanggal habis tempo)

Nama Pelanggan = Nama Depan +
 (Inisial Tengah) +
 Nama keluarga

Alamat = Jalan +
 (Apartemen) +
 Kota +
 Negara bagian +
 Zip +

		(Ekspansi Zip) + (Negara)
Telepon	=	Kode Area + Nomor lokal
Item pesanan yang Tersedia	=	Jumlah yang Dipesan + Nomor Barang + Deskripsi Barang + Ukuran + Warna + Harga + Jumlah barang
Cara Pembayaran	=	[Cek Tagihan Wesel]
Jenis Kartu Kredit	=	[Visa MasterCard American Express Tren Dunia]
Laporan Tagihan Pelanggan	=	Tanggal Sekarang + Nomor Pelanggan + Nama Pelanggan + Alamat + {Jalur Pemesanan} + (Jumlah Pembayaran Sebelumnya) + Jumlah Total Hutang + (Komentar)
Jalur Pemesanan	=	Nomor Pesanan + Tanggal Pemesanan + Jumlah Pesanan

2.3.4 Database

Database adalah kumpulan dari item data yang saling berhubungan satu dengan lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, yang tersimpan di *hardware* komputer dan dengan *software* yang digunakan untuk melakukan manipulasi data dengan tujuan tertentu. *Database* juga dapat diartikan sebagai sekumpulan data yang disusun dalam beberapa tabel saling berhubungan [9].

Manfaat *database* yaitu [9]:

1. Kecepatan dan kemudahan (*speed*)
2. Kebersamaan pemakaian (*sharability*)

3. Pemusatan kontrol data (*Control*)
4. Efisiensi ruang penyimpanan (*space*)
5. Keakuratan (*Accurancy*)
6. Ketersediaan (*availability*)
7. Keamanan (*security*)
8. Kelengkapan (*completeness*)

Istilah-istilah yang digunakan dalam *database*, yaitu [5]:

1. Atribut

Atribut adalah beberapa karakteristik dari suatu entitas, dan setiap entitas memiliki banyak atribut.

Contoh: Pasien (entitas) dapat memiliki banyak atribut, seperti nama belakang, nama depan, alamat, negara.

2. *Record*

Record adalah kumpulan item data yang memiliki kesamaan dengan entitas yang dideskripsikan.

Contoh: NIM, nama, alamat, tanggal lahir.

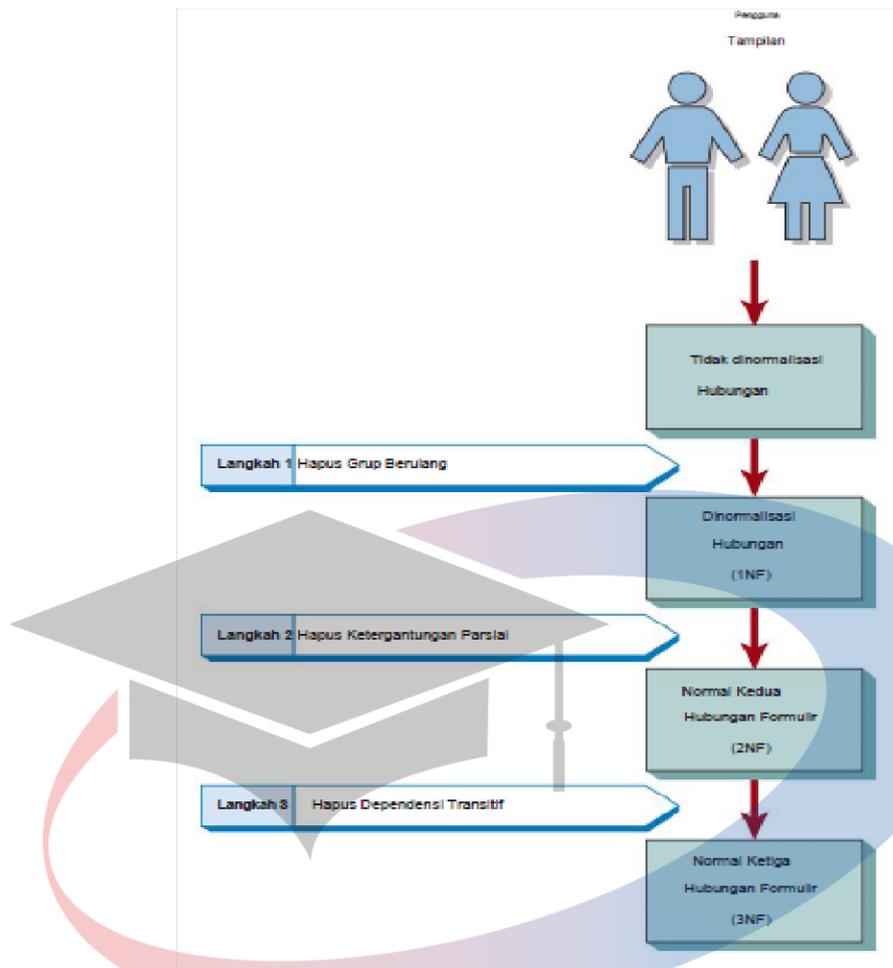
3. Kunci

Kunci adalah salah satu item data dalam catatan yang digunakan untuk mengidentifikasi catatan. Ketika kunci secara unik mengidentifikasi catatan, disebut kunci utama. Misalnya, NIM dapat menjadi kunci utama karena hanya satu nomor yang ditetapkan untuk kepemilikan nomor induk mahasiswa. Sebuah kunci disebut kunci sekunder jika tidak dapat secara unik mengidentifikasi *record*. Misalnya, alamat, jurusan.

2.3.5 Normalisasi

Normalisasi adalah transformasi tampilan pengguna yang kompleks dan penyimpanan data menjadi sekumpulan struktur data yang lebih kecil dan stabil. Selain lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah dipelihara daripada struktur lainnya. Sederhananya, normalisasi adalah proses yang menghilangkan kelompok berulang yang ditemukan dalam catatan panjang variabel [5].

Bentuk-bentuk tahapan normalisasi sebagai berikut [5]:



Gambar 2.8 Tahapan Normalisasi

Gambar 2.9 adalah tampilan pengguna untuk Perusahaan Peralatan Hidrolik Sumur Al S. Laporan menunjukkan (1) NOMOR SALESPERSON, (2) NAMA SALESPERSON, dan (3) AREA PENJUALAN. Badan laporan menunjukkan (4) NOMOR PELANGGAN dan (5) NAMA PELANGGAN. Selanjutnya adalah (6) GUDANG-NUMBER yang akan melayani pelanggan.

AI S. Yah
Perusahaan Peralatan Hidrolik
Lembah Musim Semi, Minnesota

Penjual #: 3462
 Nama: Perairan
 Area Penjualan: Barat

PELANGGAN NOMOR	PELANGGAN NAMA	GUDANG NOMOR	GUDANG LOKASI	PENJUALAN
18765	Layanan Delta	4	Fargo	13.540
18830	M. Levy dan Sons	3	Bismarck	10.600

Gambar 2.9 Laporan Penjualan Perusahaan Peralatan Hidrolik

Berikut adalah bentuk-bentuk normalisasi data, yaitu:

1. Bentuk yang tidak normal (*Unnormalization Form*)

Bentuk yang tidak normal adalah suatu kumpulan data yang akan diolah yang diperoleh dari format-format yang beraneka ragam, masih terdapat duplikasi atau pengulangan data, bisa saja tidak sempurna atau tidak lengkap, dan sesuai fakta lapangan.

Contoh bentuk yang tidak normal dapat dilihat pada tabel:

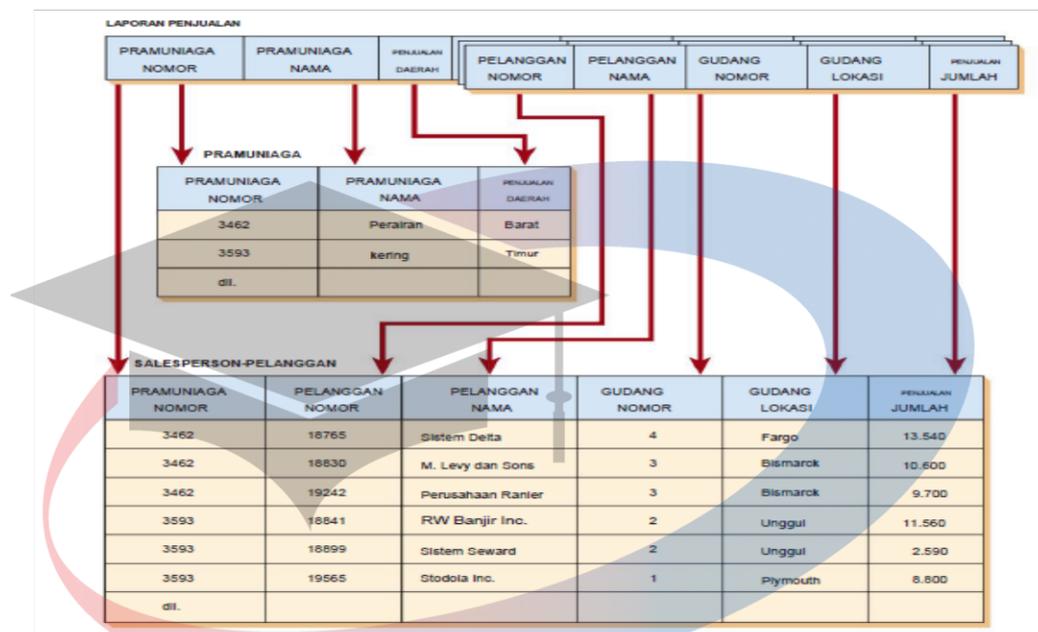
PRAMUNIAGA NOMOR	PRAMUNIAGA NAMA	PRAMUNIAGA DAERAH	PELANGGAN NOMOR	PELANGGAN NAMA	GUDANG NOMOR	GUDANG LOKASI	PRAMUNIAGA JUMLAH
3462	Perairan	Barat	18765	Delta Systems	4	Fargo	13.540
			18830	M. Levy and Sons	3	Bismarck	10.600
			19242	Ranler Company	3	Bismarck	9.700
3593	kering	Timur	18841	RW Banjir Inc.	2	Unggul	11.560
			18899	Seward Systems	2	Unggul	2.590
			19565	Stodola's Inc.	1	Plymouth	8.800
dll.							

Gambar 2.10 Tabel *Unnormalisasi*

2. Bentuk normal pertama (1st Normal Form / 1NF)

Bentuk normal pertama terpenuhi jika sebuah tabel tidak memiliki atribut yang bernilai banyak artinya setiap pertemuan baris dan kolom hanya berisikan satu nilai. Bentuk ini berfungsi untuk menghilangkan perulangan grup, dimana relasi yang tidak dinormalisasikan akan dipecah menjadi dua relasi terpisah.

Contoh bentuk 1NF dapat dilihat pada gambar:

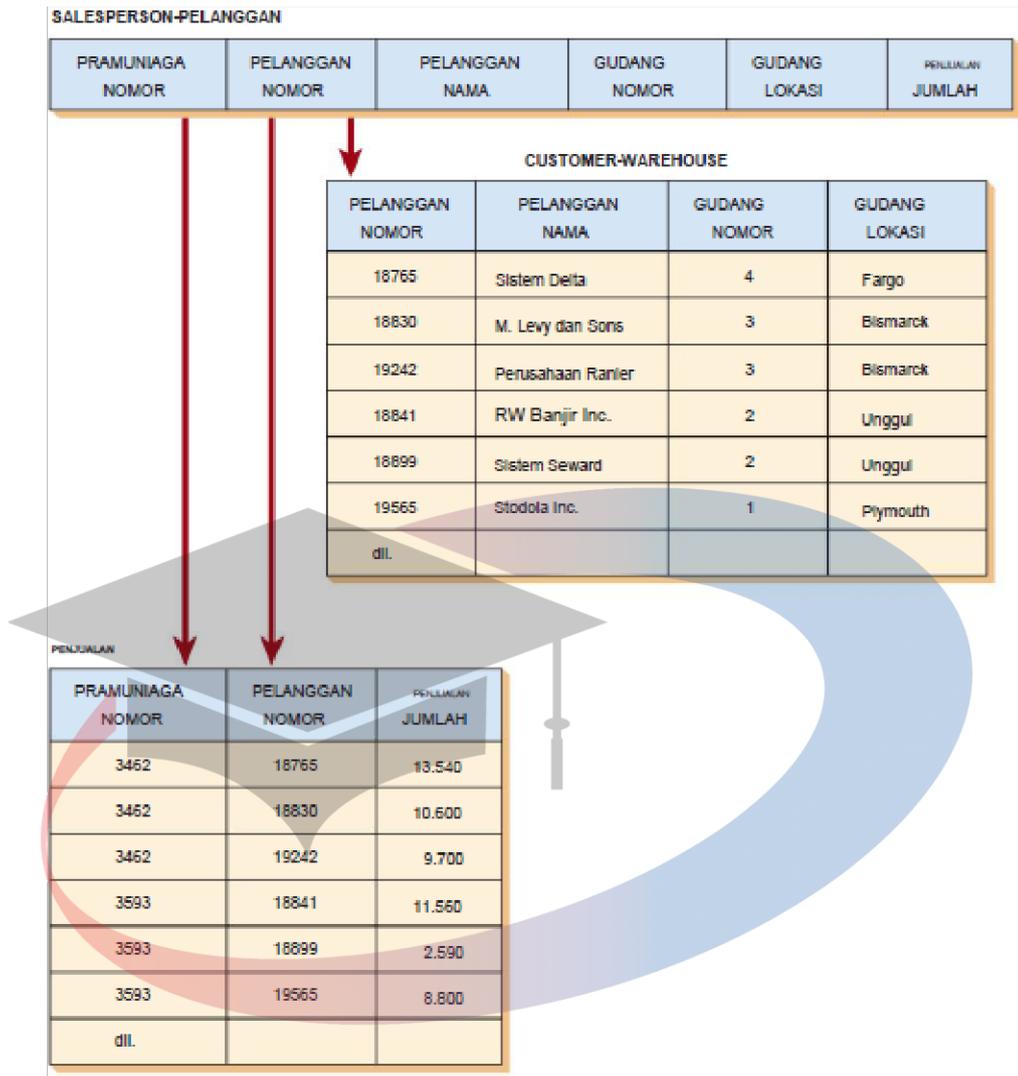


Gambar 2.11 Bentuk Normal Pertama

3. Bentuk normal kedua (2NF)

Bentuk normal 2NF dikatakan terpenuhi jika telah memenuhi bentuk 1NF, dan semua atributnya akan tergantung secara fungsional pada kunci utama. Dalam tahapan ini akan dilakukan proses menghapus semua atribut yang bergantung sebagian dan menempatkannya ke dalam relasi lain.

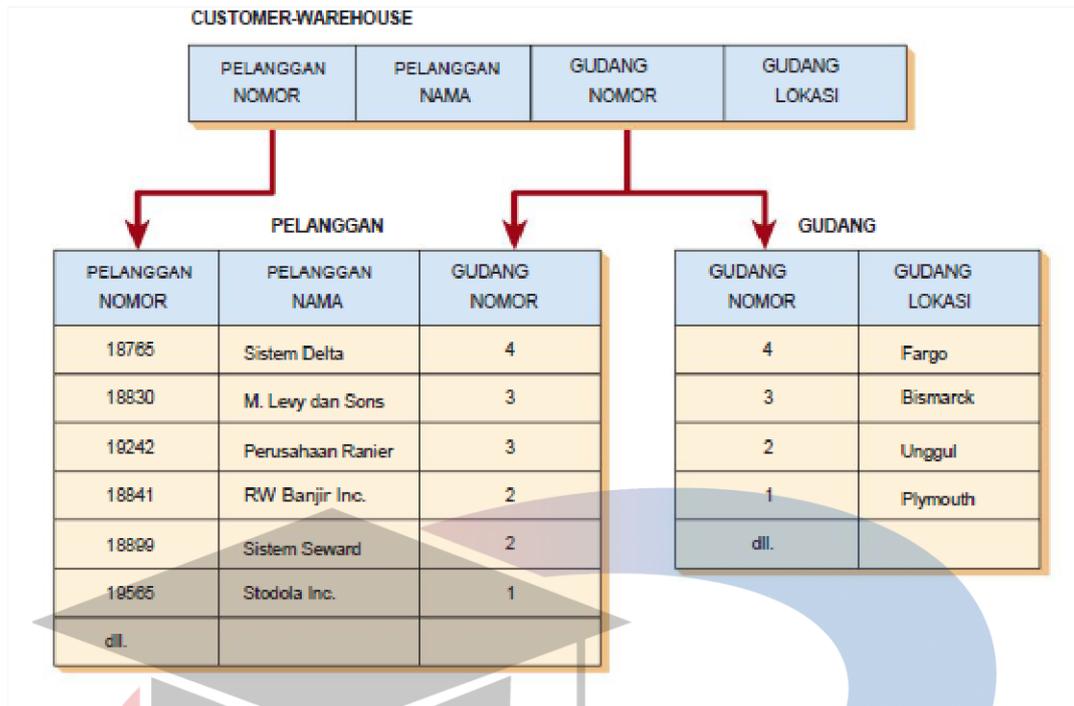
Contoh bentuk 2NF dapat dilihat pada gambar:



Gambar 2.12 Bentuk Normal Kedua

4. Bentuk normal ketiga (3NF)

Bentuk normal 3NF jika semua atribut bukan kunci sepenuhnya bergantung secara fungsional pada kunci utama dan tidak ada ketergantungan transitif (bukan kunci). Hal ini dapat dilihat dari gambar di bawah ini:



Gambar 2.13 Bentuk Normal Ketiga

2.4 Website

Website merupakan sebuah sistem dimana informasi disediakan dalam bentuk teks, gambar, suara, dan lain-lain yang dipresentasikan dalam bentuk *hypertext* yang dapat diakses perangkat lunak yang disebut *browser* [10].

Fungsi *website* secara umum, yaitu [10]:

1. Fungsi komunikasi
2. Fungsi informasi
3. Fungsi hiburan
4. Fungsi transaksi

Jenis-jenis *website*, yaitu [3]:

1. *Website* statis, yakni *website* yang informasinya tidak diperbaharui dalam waktu tertentu, sehingga informasi dari waktu ke waktu tetap sama. Contohnya profil perusahaan.
2. *Website* dinamis, yakni *website* yang informasinya selalu diperbaharui oleh pengelola *website*. Contohnya <http://www.anneahira.com/>, dan lain-lain.

2.5 Jasa Kecantikan

Jasa merupakan setiap kegiatan ekonomi yang ditawarkan oleh orang kepada orang lain, yang pada dasarnya tidak berwujud dan tidak ada kepemilikan apapun tetapi bisa dinikmati manfaatnya [11].

Kecantikan berasal dari kata cantik yang berarti bagus, elok, molek, indah, sedap dipandang. Sedangkan kecantikan adalah keindahan, kemolekan, keelokan baik tentang wajah atau bentuk tubuh [12].

Kecantikan pada umumnya diklasifikasikan ke dalam tiga kelompok, yaitu [12]:

1. Kecantikan luar (*outer beauty*), misalnya: kulit putih, tubuh yang langsing, bersih, wajah.
2. Kecantikan dari (*inner beauty*), misalnya: jiwa dan hati, akal dan pikiran, dan kepribadian.

