

## BAB II TINJAUAN PUSTAKA

### 2.1 *Internet*

*Internet (Interconnected Network)* adalah komunikasi global yang terdiri dari berbagai hubungan komputer dan jaringan komputer secara global. Komputer dan jaringan dapat terhubung secara langsung dan tidak langsung ke beberapa jalur utama yang disebut *internet* backbone dan dibedakan antara satu dengan lainnya dengan nama yang unik, disebut sebagai alamat IP 32 bit (misalnya 202.155.4.230). Komputer dan jaringan dari berbagai platform yang memiliki perbedaan dan (Unix, Linux, Windows, Mac, dll) bertukar informasi dengan protokol standar dikenal dengan TCP/IP (Transmission Control Protocol/*Internet* Protocol). [3]

Fungsi *internet* dengan seiring perkembangan waktu semakin lebih sedikit, yang artinya hampir tidak bisa disebutkan secara spesifik tetapi secara global, *internet* dapat digolongkan menjadi beberapa fungsi, yaitu [3]:

1. Fungsi *internet* sebagai media komunikasi adalah kegunaan dari *internet* yang sangat banyak digunakan oleh masyarakat agar dapat berkomunikasi satu dengan yang lainnya di seluruh dunia tanpa mengenal batas atau jarak.
2. Fungsi *internet* sebagai media pertukaran data adalah kegunaan dari *internet* dengan memanfaatkan *internet* sebagai media pertukaran data menggunakan email, ftp, google drive atau aplikasi seperti whatsapp, instagram dan lain sebagainya yang bisa digunakan untuk bertukar informasi kapan dan di mana pun.
3. Fungsi *internet* sebagai media mencari informasi atau data, dengan perkembangan *internet* yang cepat membuat *www* menjadi salah satu sumber informasi penting dan tepat.
4. Fungsi *internet* sebagai fungsi komunitas, *internet* membentuk kelompok masyarakat yang beranggotakan pengguna *internet* dari berbagai dunia. Komunitas dari pengguna *internet* dapat melakukan berbagai hal seperti berkomunikasi, berbagi informasi, berbelanja *online*, melakukan transaksi bisnis *online* dan sebagainya.
5. Fungsi *internet* sebagai media pendidikan. Komunikasi melalui *internet* dapat dilakukan secara pribadi (misalnya *chatting*) atau secara bersama-sama, yang

dikenal *one to many communication* (misalnya *zoom*). *Internet* sebagai media pendidikan mampu menghadapi karakteristik yang khas, yaitu sebagai media antar pengguna personal dan banyak pengguna; bersifat interaktif, memungkinkan komunikasi secara sinkron maupun asinkron.

## 2.2 *E-commerce*

*E-commerce (Electronic Commerce)* adalah penyebaran, penjualan, pembelian barang ataupun jasa dengan menggunakan *internet*, televisi atau jaringan komputer lainnya sebagai sarana elektronik. [4] *E-commerce* merupakan bagian dari aplikasi bisnis, dimana lingkup aplikasi bisnis lebih luas, tidak hanya perdagangan tetapi meliputi juga penggabungan antara mitra bisnis, pelayanan nasabah, lowongan pekerjaan dan lain sebagainya.

*E-commerce* memiliki berbagai media sebagai platform yang bisa digunakan. Media Komunikasi *e-commerce* yang pertama adalah website. Website *e-commerce* yaitu perdagangan elektronik *internet* yang dimana terdapat transaksi penjualan dan pembelian barang maupun jasa menggunakan situs *internet*. Media sosial adalah media komunikasi yang kedua yang bisa digunakan. Peran media sosial sangat penting dalam pemasaran barang atau jasa, penjual bisa mempromosikan dan menjual produk kepada pelanggan dengan jangkauan yang luas. Aplikasi *mobile* adalah media komunikasi ketiga yang bisa digunakan. Dengan menggunakan aplikasi *mobile*, pelanggan tidak harus menghafal URL yang panjang untuk masuk ke akun pelanggan beberapa kali. Aplikasi *mobile* juga lebih cepat dalam penggunaannya dan memiliki keamanan lebih baik dari pada situs seluler. [5]

Berbagai keuntungan dapat diperoleh dari *e-commerce*, yaitu menjual produk maupun jasa *online* tanpa harus membangun toko secara fisik yang selayaknya dibangun oleh penjual produk offline sebagai tempat usaha. Proses penjualan yang dilakukan hanya dengan menggunakan *internet*, maka penjual sudah bisa memasarkan produk kepada pelanggan kapan dan di mana pun. *E-commerce* juga membuat penjual dan pelanggan mudah dalam berkomunikasi dan tidak perlu biaya yang mahal untuk melakukan promosi, dengan hanya terhubung ke jaringan *internet* maka penjual dapat memasarkan produk atau jasa ke jangkauan yang lebih luas kepada masyarakat. [5]

Pada dasarnya ada 4 (empat) jenis relasi dalam dunia bisnis yang biasa dijalin oleh perusahaan, yaitu [6]:

1. Relasi dengan pemasok (*supplier*);
2. Relasi dengan distributor;
3. Relasi dengan rekan (*partner*);
4. Relasi dengan pelanggan (*customer*).

### 2.3 *Mobile Commerce*

*M-commerce* (perdagangan seluler) adalah proses menjual dan membeli barang atau jasa melalui perangkat seluler. *M-commerce* akan terus berkembang di masa mendatang. Teknologi memiliki karakteristik dan batasannya, tetapi memiliki peluang besar untuk bisnis karena pertumbuhan penjualan dan model bisnis baru terus melampaui teknologi lainnya. Prospek bisnis nyata dalam *m-commerce* berasal dari peluang untuk mengembangkan model bisnis baru dan mengintegrasikan teknologi baru di mana inovasi dan kreativitas sangat dihargai. [7]

*M-commerce*, atau perdagangan seluler, istilah yang berasal dari perdagangan elektronik dengan fokus pada kemampuan untuk membeli, menjual, mengiklankan, dan melakukan operasi bisnis saat bepergian. Teknologi yang memungkinkan proses seperti itu terus berkembang dan berubah. *M-commerce* sekarang banyak diasosiasikan dengan *smartphone* dan tablet. Bisnis telah mengamati peluang dalam *m-commerce* dalam teknologi yang dapat dikenakan seperti jam tangan pintar dan kacamata pintar. Oleh karena itu, kriteria utama dimana suatu proses dapat dikategorikan sebagai *m-commerce* adalah kemampuan untuk menggunakan perangkat nirkabel untuk transaksi bisnis saat dalam perjalanan. [7]

*M-commerce* memiliki karakteristik yang membuatnya sangat berbeda dengan *e-commerce* yang menggunakan desktop dan laptop [7]:

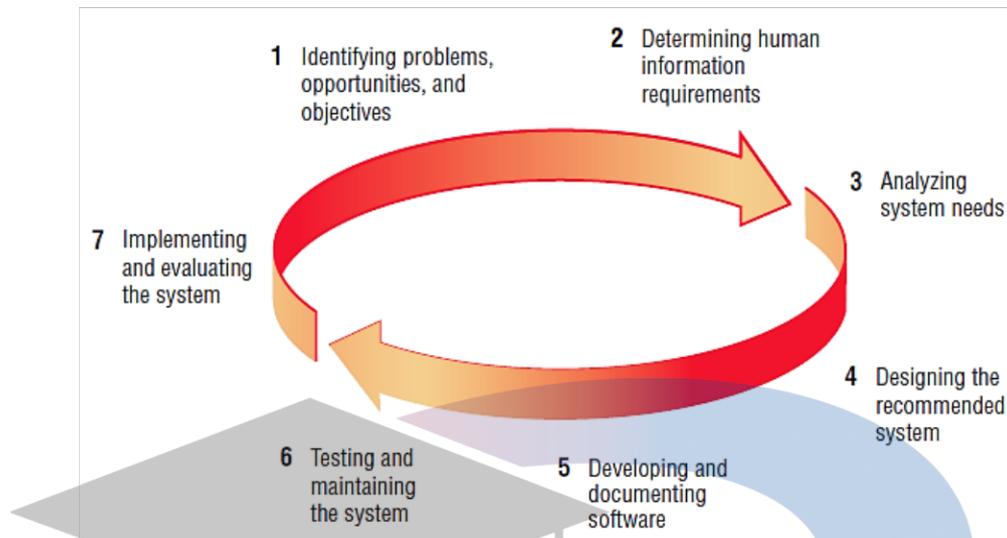
1. Perangkat seluler cenderung memiliki ruang terbatas: *smartphone* atau tablet biasa akan memiliki ruang penyimpanan kurang dari 10% dibandingkan dengan desktop dan laptop.
2. Perangkat seluler juga memiliki tampilan layar yang jauh lebih kecil untuk memudahkan mobilitas. Ini berarti desain situs web, aplikasi, dan aplikasi untuk *m-commerce* perlu mempertimbangkan apa saja persyaratan penting yang

diperlukan untuk menyesuaikan dengan tampilan terbatas ini. Perangkat seluler perlu beroperasi dengan baterai selama berjam-jam. Fitur ini penting jika perangkat dianggap benar-benar *mobile*. Penggunaan perangkat seluler yang lama dan terus menerus akan menguras baterai.

3. Lalu lintas *internet* di perangkat seluler perlu mempertimbangkan bahwa banyak perangkat seluler akan menggunakan jaringan seluler. Baik perangkat dan jaringan tidak dapat menangani lalu lintas yang luas meskipun generasi baru tablet dan ponsel cerdas telah dirancang untuk menangani lalu lintas *Internet* yang lebih cepat dan lebih baik dengan kecepatan sekarang mencapai 5 GB atau 100 GB per detik.
4. Sistem operasi pada perangkat seluler telah dirancang untuk tidak berukuran besar dan mengurangi penggunaan prosesor dan RAM secara ekstensif. Dengan demikian biaya alat lebih rendah daripada alat desktop atau laptop.
5. Perangkat seluler hadir dengan fitur ekstensif yang tidak biasa ditemukan di perangkat desktop atau laptop. Fitur termasuk kamera resolusi tinggi, Sistem Pemosisian Global (GPS), Komunikasi Nirkabel Jarak Dekat (NFC), pesan SMS, panggilan, dan akses ke perpustakaan ekstensif untuk unduhan aplikasi yang dikenal sebagai Aplikasi.

#### 2.4 *Systems Development Life Cycle (SDLC)*

*Systems Development Life Cycle (SDLC)* adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik. Tahapan *SDLC* terdiri dari 7 tahap, yaitu sebagai berikut [8]:



Gambar 2. 1 *The seven phases of the systems development life cycle (SDLC).*

1. Mengidentifikasi masalah, peluang dan tujuan

Dalam tahap pertama siklus hidup pengembangan sistem, penganalisis mengidentifikasi masalah, peluang, dan tujuan yang akan dicapai. Penganalisis melihat secara jujur apa yang terjadi dalam bisnis. Kemudian, bersama dengan anggota organisasi lain, penganalisis menentukan dengan tepat masalah-masalah tersebut. Peluang adalah situasi yang di mana penganalisis yakin bahwa melalui penggunaan sistem informasi terkomputerisasi peningkatan dapat dilakukan. Mencapai peluang memungkinkan bisnis untuk mendapatkan keunggulan kompetitif atau menetapkan standar industri.

Mengidentifikasi tujuan juga merupakan komponen penting dari fase pertama. Penganalisis harus menemukan apa yang baik untuk dilakukan dalam bisnis tersebut. Kemudian penganalisis akan dapat melihat beberapa aspek dari aplikasi sistem informasi dapat membantu bisnis mencapai tujuannya dengan menangani masalah atau peluang khusus. Orang-orang yang terlibat dalam fase pertama adalah pengguna, penganalisis dan manajer sistem yang mengoordinasikan proyek. Kegiatan pada tahap ini terdiri dari wawancara manajemen pengguna, meringkas pengetahuan yang diperoleh, memperkirakan ruang lingkup proyek, dan mendokumentasikan hasilnya. *Output* dari tahap ini adalah laporan kelayakan yang berisi definisi masalah dan ringkasan tujuan.

2. Menentukan syarat-syarat informasi

Dalam tahap selanjutnya, penganalisis menentukan memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Penganalisis akan menggunakan metode interaktif seperti wawancara, pengambilan sampel dan investigasi hard data, dan kuesioner, bersama dengan metode yang tidak mengganggu, seperti mengamati perilaku pengambil keputusan dan lingkungan kantor dan metode yang mencakup semua, seperti pembuatan prototipe.

Dalam tahap kebutuhan informasi *SDLC*, penganalisis berusaha untuk memahami informasi apa yang dibutuhkan pengguna untuk melakukan pekerjaan penganalisis. Pada tahap ini penganalisis sedang mempelajari bagaimana membuat sistem berguna bagi orang-orang yang terlibat. Orang-orang yang terlibat dalam fase ini adalah penganalisis dan pengguna, biasanya manajer operasi dan pekerja operasi. Penganalisis sistem perlu mengetahui rincian fungsi sistem saat ini: siapa (orang-orang yang terlibat), apa (aktivitas bisnis), di mana (lingkungan tempat pekerjaan berlangsung), kapan (waktu), dan bagaimana (bagaimana prosedur saat ini dilakukan) dari bisnis yang diteliti. Penganalisis kemudian harus bertanya mengapa bisnis menggunakan sistem saat ini.

### 3. Menganalisis kebutuhan sistem

Dalam tahap selanjutnya penganalisis sistem melibatkan analisis kebutuhan sistem. Alat dan teknik khusus akan membantu penganalisis membuat penentuan persyaratan. Alat seperti diagram aliran data (*DFD*) untuk memetakan *input*, proses, dan *output* dari fungsi bisnis, atau diagram aktivitas atau diagram urutan untuk menunjukkan urutan peristiwa, menggambarkan sistem dalam bentuk grafik yang terstruktur. Dari diagram aliran data, dikembangkan suatu kamus data berisikan daftar seluruh item data yang digunakan dalam sistem, berikut spesifiknya apakah berupa *alphanumeric* atau teks, serta berapa banyak spasi yang dibutuhkan saat dicetak. Selama fase ini penganalisis sistem juga menganalisis keputusan terstruktur yang dibuat. Keputusan terstruktur adalah keputusan di mana kondisi, alternatif kondisi, tindakan, dan aturan tindakan dapat ditentukan. Ada tiga metode utama untuk menganalisis keputusan terstruktur: bahasa Inggris terstruktur, tabel keputusan, dan pohon keputusan. Pada titik di *SDLC* ini, penganalisis sistem menyiapkan proposal sistem yang merangkum apa yang telah ditemukan tentang pengguna dan kegunaan sistem saat ini, memberikan analisis

biaya-manfaat dari alternatif dan membuat rekomendasi tentang apa (jika ada) yang harus dilakukan. Jika salah satu rekomendasi dapat diterima oleh manajemen, penganalisis melanjutkan jalannya. Setiap masalah sistem itu unik dan tidak pernah ada hanya satu solusi yang benar. Solusi yang dirumuskan bergantung pada kualitas individu dan pelatihan profesional dari setiap penganalisis dan interaksi penganalisis dengan pengguna dalam konteks lingkungan kerja.

#### 4. Merancang sistem yang direkomendasikan

Dalam tahap desain *SDLC*, penganalisis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk menyelesaikan desain logis dari sistem informasi. Penganalisis merancang prosedur bagi pengguna untuk membantu pengguna memasukkan data secara akurat, sehingga data yang masuk ke sistem informasi benar. Selain itu, penganalisis menyediakan bagi pengguna untuk melengkapi masukan efektif ke sistem informasi dengan menggunakan teknik formulir yang baik dan desain halaman web atau layar. Bagian dari desain logis dari sistem informasi adalah merancang *HCI*. Antarmuka menghubungkan pengguna dengan sistem dan karenanya sangat penting. Antarmuka pengguna dirancang dengan bantuan pengguna untuk memastikan bahwa sistem dapat didengar, terbaca, dan aman, serta menarik dan menyenangkan untuk digunakan. Contoh antarmuka pengguna fisik termasuk keyboard (untuk mengetik pertanyaan dan jawaban), menu di layar (untuk mendapatkan perintah pengguna) dan berbagai antarmuka pengguna grafis (*GUI*) yang menggunakan mouse atau layar sentuh.

Fase desain juga mencakup mendesain database yang akan menyimpan banyak data yang dibutuhkan oleh pembuat keputusan dalam organisasi. Dalam fase ini, penganalisis juga bekerja dengan pengguna untuk mendesain *output* (baik di layar maupun dicetak) yang memenuhi kebutuhan informasi pengguna. Akhirnya, penganalisis harus merancang kontrol dan prosedur cadangan untuk melindungi sistem dan data dan untuk menghasilkan paket spesifikasi program untuk pemrogram. Setiap paket harus berisi tata letak *input* dan *output*, spesifikasi file, dan detail pemrosesan; itu juga dapat mencakup pohon keputusan atau tabel, *UML* atau diagram aliran data, dan nama dan fungsi dari kode yang telah ditulis

sebelumnya baik yang ditulis sendiri atau menggunakan kode atau pustaka kelas lainnya.

5. Mengembangkan dan mendokumentasikan perangkat lunak

Dalam tahap kelima *SDLC*, penganalisis bekerja dengan pemrogram untuk mengembangkan perangkat lunak awal yang diperlukan. Dokumentasi memberi tahu pengguna cara menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi masalah perangkat lunak. Pemrogram memiliki peran kunci dalam tahap ini karena pemrogram merancang, membuat kode, dan menghapus kesalahan sintaksis dari program komputer. Untuk memastikan kualitas, programmer dapat melakukan desain atau panduan kode, menjelaskan bagian kompleks dari program kepada tim programmer lain.

6. Menguji dan mempertahankan sistem

Sebelum sistem informasi dapat digunakan, maka harus dilakukan pengujian. Beberapa pengujian diselesaikan oleh pemrogram sendiri, beberapa di antaranya oleh penganalisis sistem bersama dengan pemrogram. Pemeliharaan sistem dan dokumentasinya dimulai pada tahap ini dan dilakukan secara rutin sepanjang umur sistem informasi. Sebagian besar pekerjaan rutin programmer terdiri dari pemeliharaan, dan bisnis menghabiskan banyak uang untuk pemeliharaan. Beberapa pemeliharaan, seperti pembaruan program, dapat dilakukan secara otomatis melalui situs vendor di *Web*. Banyak dari prosedur sistematis yang digunakan penganalisis di seluruh *SDLC* dapat membantu memastikan bahwa pemeliharaan dijaga seminimal mungkin.

7. Mengimplementasikan dan mengevaluasi vendor

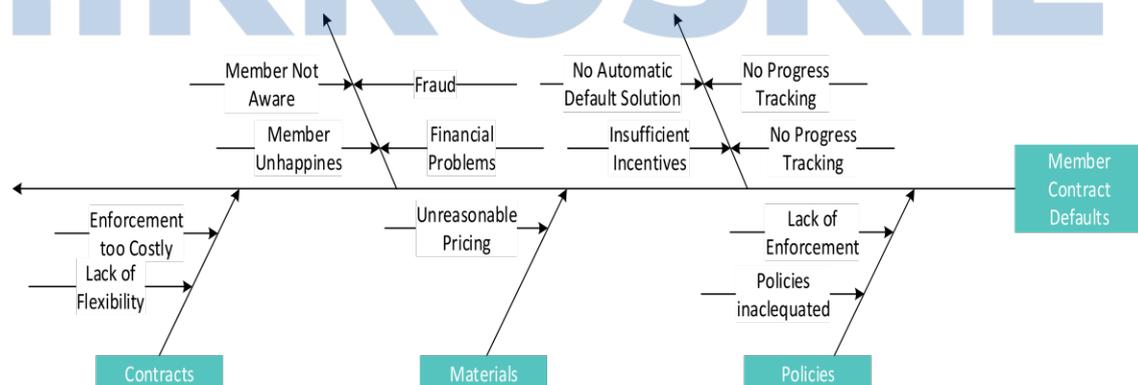
Pada tahap terakhir pengembangan sistem ini, penganalisis membantu mengimplementasikan sistem informasi. tahap ini melibatkan pelatihan pengguna untuk menangani sistem. Vendor melakukan beberapa pelatihan, tetapi pengawasan pelatihan adalah tanggung jawab penganalisis sistem. Selain itu, penganalisis perlu merencanakan konversi yang lancar dari sistem lama ke yang baru. Proses ini termasuk mengonversi file dari format lama ke format baru, atau membangun database, memasang peralatan, dan membawa sistem baru ke dalam produksi.

## 2.5 Teknik Analisis dan Perancangan Sistem

### 2.5.1 Fishbone Diagram

*Fishbone Diagram* juga dikenal sebagai Diagram Ishikawa merupakan proses pemecahan masalah atau kebenaran yang diketahui dan memberikan cara yang teratur untuk mengamati hasil dan alasan yang menjelaskan efek tersebut. Karena maksud dari diagram Fishbone, ini dapat didelegasikan sebagai diagram sebab-akibat oleh Watson pada tahun 2004. Diagram Fishbone (Ishikawa) terutama mewakili model presentasi sugestif untuk korelasi antara suatu peristiwa (efek) dan kelipatannya penyebab. Struktur yang disediakan oleh diagram membantu anggota tim berpikir dengan cara yang sangat sistematis. Diagram tulang ikan secara efektif digunakan untuk mencapai akar penyebab masalah. [9]

Penggambaran diagram tulang ikan dimulai dengan nama masalah yang akan digambar pada sebelah kanan diagram (atau kepala ikan). Penyebab dari masalah tersebut kemudian digambar sebagai tulang ikan sepanjang tulang belakang, masing-masing sebab masalah digambar dalam sebuah anak panah yang mengarah ke tulang belakang. Biasanya, "tulang" ini diberi label sebagai empat kategori dasar yaitu bahan (*materials*), mesin (*machines*), tenaga kerja (*manpower*) dan metode (*methods*) yang disebut sebagai *4Ms*. Nama lain dapat digunakan untuk menyesuaikan dengan masalah yang dihadapi. Kategori alternatif atau tambahan termasuk tempat (*place*), prosedur (*procedures*), kebijakan (*policies*), dan manusia (*people*), yang disebut *4Ps* atau lingkungan (*surroundings*), pemasok (*suppliers*), sistem (*systems*), dan keterampilan (*skills*), yang disebut sebagai *4Ss* [10].



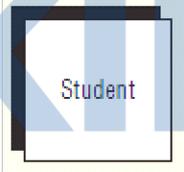
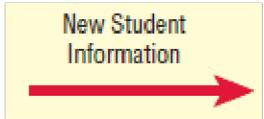
Gambar 2. 2 Diagram Fishbone

### 2.5.2 Data Flow Diagram (DFD)

*Data Flow Diagram (DFD)* merupakan diagram aliran data yang menggambarkan gambaran umum seluas mungkin dari *input*, proses dan *output* sistem. Penganalisis sistem perlu memanfaatkan kebebasan konseptual yang ada pada diagram aliran data, secara grafis mencirikan proses dan aliran data dalam sistem bisnis [8].

Ketika penganalisis sistem mencoba untuk memahami kebutuhan informasi dari pengguna, penganalisis harus dapat membuat konsep bagaimana data bergerak melalui organisasi, proses atau transformasi yang dilakukan data dan apa *output*-nya. Melalui teknik analisis terstruktur yang disebut diagram aliran data (*DFD*), penganalisis sistem dapat mengumpulkan representasi grafis dari proses data di seluruh organisasi. Dengan menggunakan kombinasi hanya empat simbol, penganalisis sistem dapat membuat penggambaran bergambar proses yang pada akhirnya akan memberikan dokumentasi sistem yang solid. Empat simbol dasar digunakan untuk memetakan pergerakan data pada diagram aliran data adalah [8]:

Tabel 2. 1 Tabel Simbol Dasar *DFD*

Gambar	Pengertian	Fungsi	Contoh
	Entitas	Menggambarkan suatu entitas eksternal (bagian lain, perusahaan, seseorang atau mesin) yang dapat mengirim dan menerima data dari sistem	
	<i>Data Flow</i>	Menunjukkan perpindahan data dari satu titik ke titik lain,	

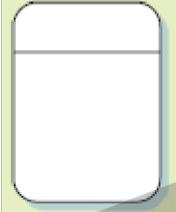
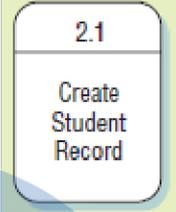
		dengan kepala tanda panah mengarah ke tujuan data.	
	Proses	Menunjukkan adanya perubahan proses.	
	Data Store	Menunjukkan penyimpanan data yang memungkinkan penambahan dan perolehan data.	

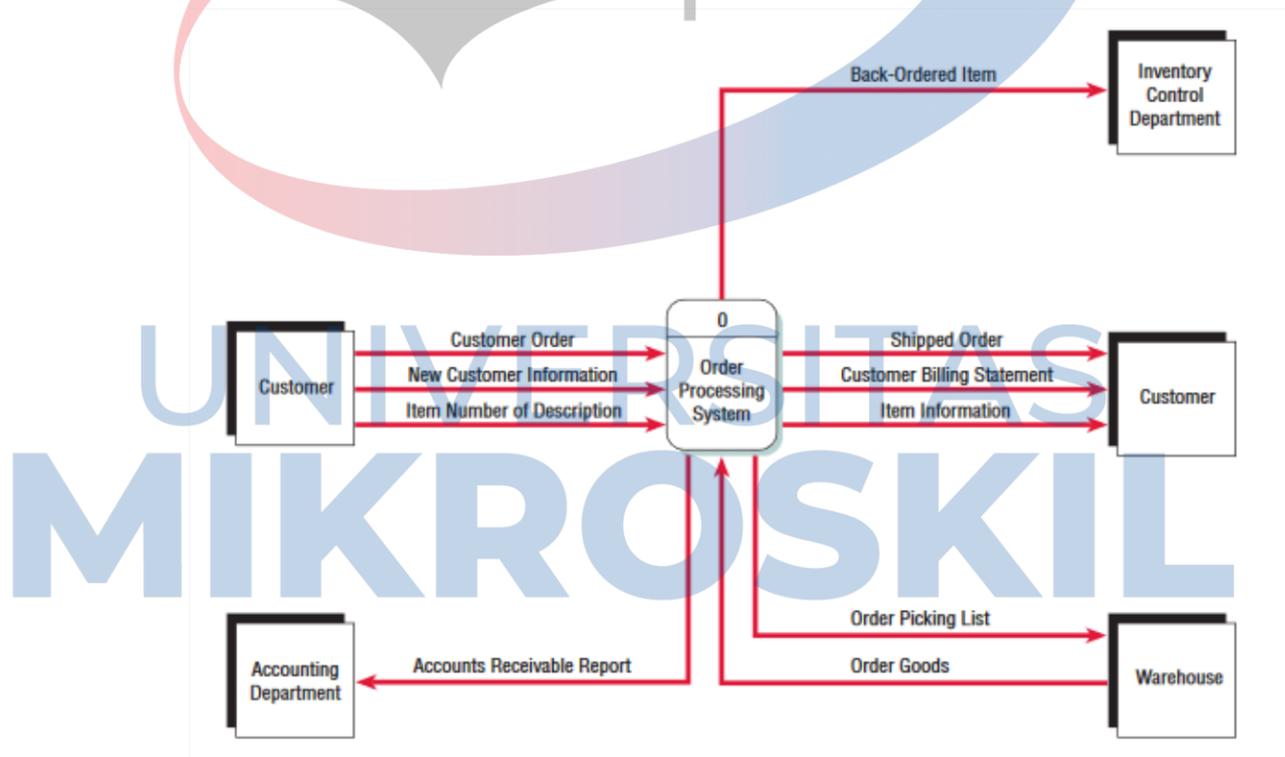
Diagram aliran data dapat dan harus digambar secara sistematis. Pertama, penganalisis sistem perlu mengkonseptualisasikan aliran data dari perspektif *top-down*. Untuk memulai diagram aliran data, masukkan narasi (atau cerita) sistem organisasi ke dalam daftar dengan empat kategori entitas eksternal, aliran data, proses, dan penyimpanan data. Daftar tersebut pada gilirannya membantu menentukan batas-batas sistem yang akan digambarkan. Setelah daftar dasar elemen data telah disusun, mulailah menggambar diagram konteks. Berikut beberapa aturan dasar yang harus diikuti [8]:

1. Diagram aliran data harus memiliki setidaknya satu proses dan tidak boleh ada objek yang berdiri sendiri yang terhubung.
2. Sebuah proses harus menerima setidaknya satu aliran data yang masuk ke dalam proses dan membuat setidaknya satu aliran data yang keluar dari proses.
3. Penyimpanan data harus terhubung ke setidaknya satu proses.
4. Entitas eksternal tidak boleh terhubung satu sama lain. Meskipun berkomunikasi secara independen, komunikasi tersebut bukanlah bagian dari sistem yang dirancang menggunakan *DFD*.

Langkah-langkah menggambar diagram aliran data [8]:

1. Merancang Diagram Konteks (*Context Diagram*)

Dengan pendekatan atas-bawah untuk membuat diagram pergerakan data, diagram berpindah dari umum ke yang lebih spesifik. Meskipun diagram pertama membantu penganalisis sistem memahami pergerakan data dasar, sifat umumnya dan membatasi kegunaannya. Diagram konteks awal harus berupa gambaran umum, yang mencakup masukan dasar, sistem umum, dan keluaran. Diagram konteks akan menjadi diagram yang paling umum, dengan pandangan sekilas tentang pergerakan data dalam sistem dan konseptualisasi sistem seluas mungkin. Diagram konteks adalah level tertinggi dalam diagram aliran data dan hanya berisi satu proses, mewakili keseluruhan sistem. Proses tersebut diberi angka nol. Berikut contoh penggambaran diagram konteks.

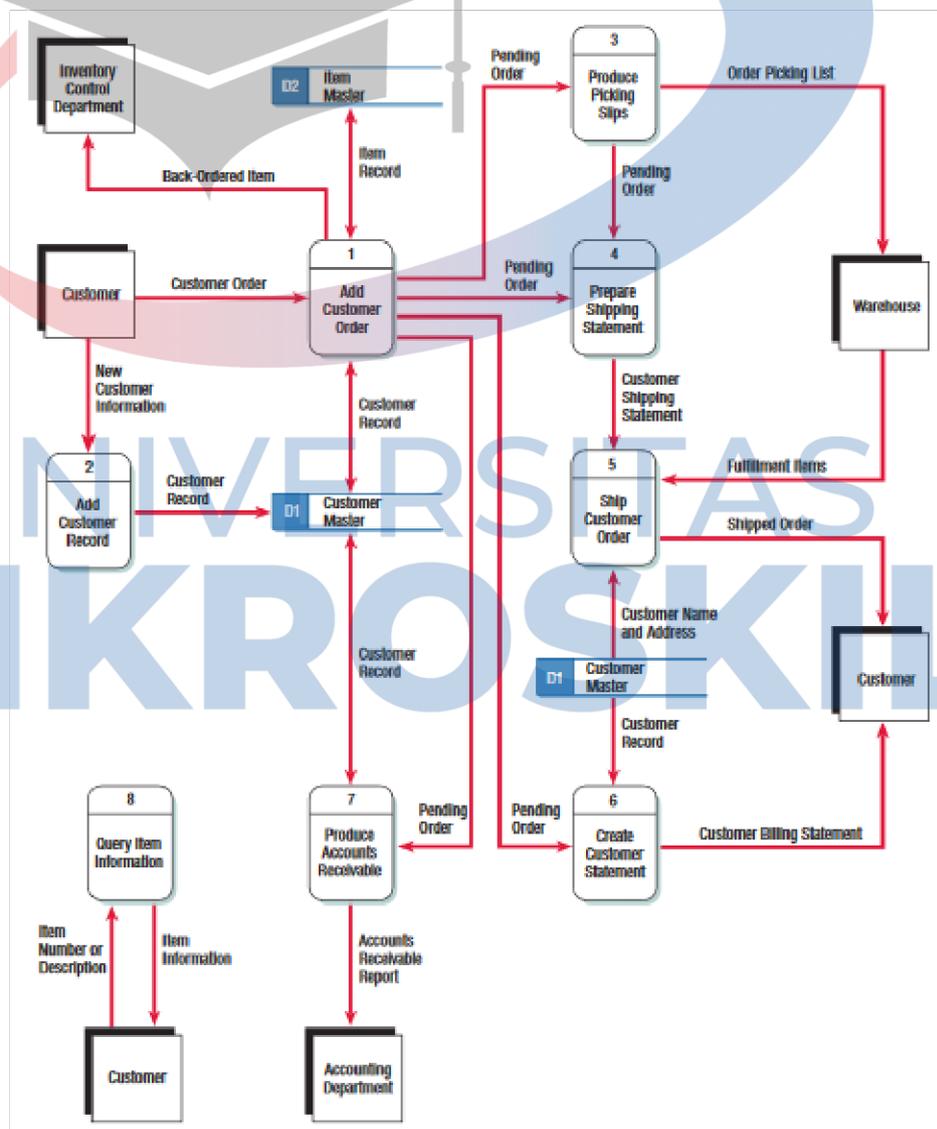


Gambar 2. 3 Diagram Konteks

2. Merancang Diagram 0 (*The Next Level*)

Selanjutnya, kembali ke daftar aktivitas dan buat daftar baru sebanyak proses dan penyimpanan data yang dapat ditemukan. Pengguna terlebih dahulu membuat

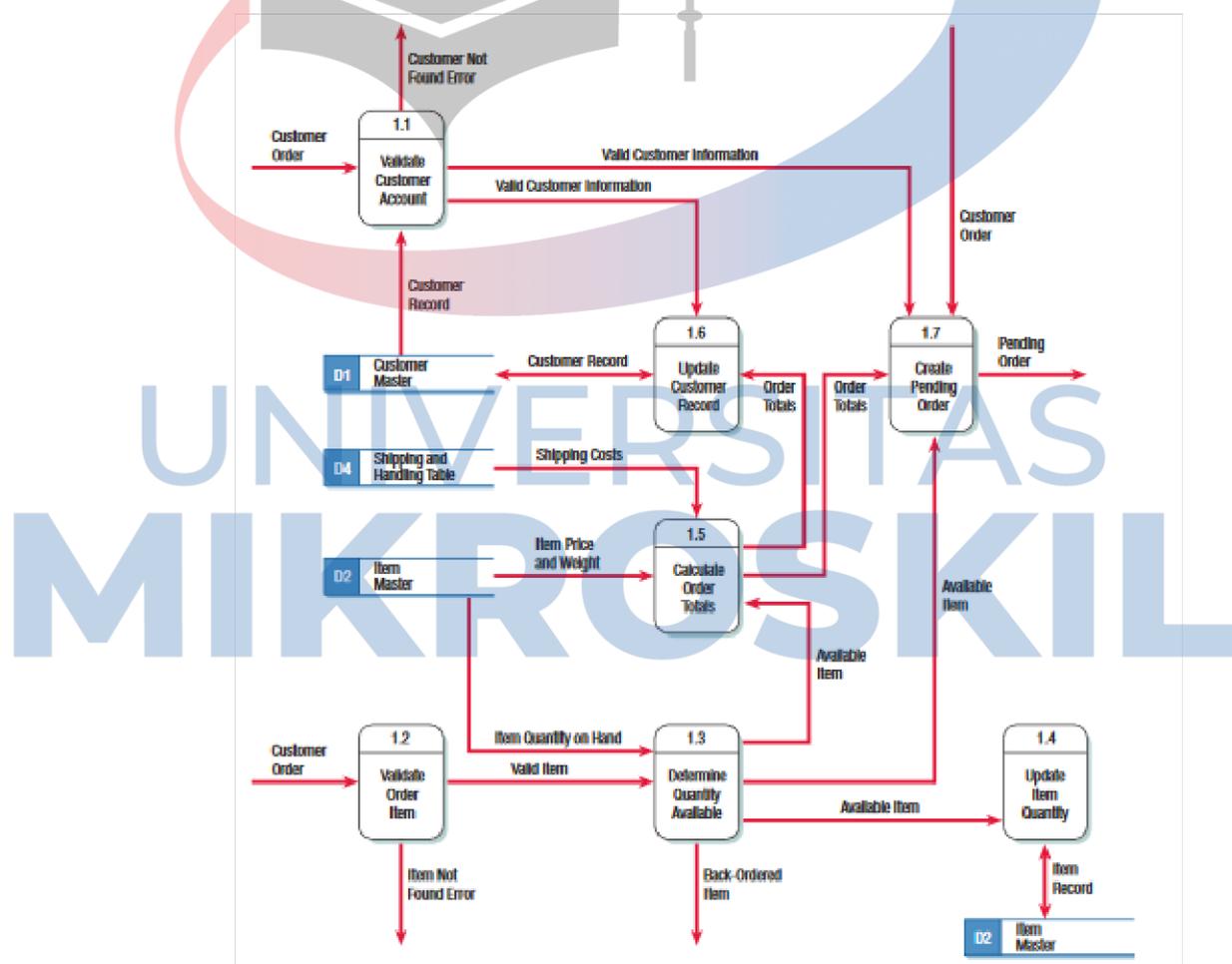
daftar. Jika pengguna merasa memiliki informasi yang cukup, buatlah diagram level 0 seperti yang ditemukan pada Gambar 2.4. Masukan dan keluaran yang ditetapkan dalam diagram konteks tetap konstan. Sisa diagram konteks dikembangkan ke dalam gambaran terperinci yang melibatkan tiga sampai sembilan proses, serta menunjukkan penyimpanan data dan aliran data baru pada level yang lebih rendah. Diagram 0 adalah pengembangan diagram konteks dan bisa mencakup sampai sembilan proses. Memasukkan lebih banyak proses pada level ini membuat diagram menjadi rumit. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut kiri atas diagram dan mengarah ke sudut kanan bawah diagram.



Gambar 2. 4 Diagram 0

### 3. Merancang Diagram Anak (*More Detailed Levels*)

Setiap proses pada Diagram 0 pada gilirannya dapat dibagi untuk membuat diagram anak yang lebih detail. Proses pada Diagram 0 yang dikembangkan disebut proses induk dan diagram yang menghasilkan disebut diagram anak. Aturan utama untuk membuat diagram anak, penyeimbangan vertikal, menyatakan bahwa diagram anak tidak dapat menghasilkan keluaran atau menerima masukan dari induk, proses juga tidak menghasilkan atau menerima. Semua aliran data masuk atau keluar dari proses induk harus ditampilkan mengalir ke dalam atau keluar dari diagram anak.



Gambar 2. 5 Diagram Anak

### 2.5.3 Kamus Data

Kamus data merupakan aplikasi khusus dari jenis kamus yang digunakan sebagai referensi dalam kehidupan sehari-hari. Kamus data adalah karya referensi dari data tentang data (yaitu, metadata), yang disusun oleh penganalisis sistem untuk memandu pengguna melalui analisis dan desain [8]. Selain menyediakan dokumentasi dan menghilangkan redundansi, kamus data digunakan untuk [8]:

1. Validasi diagram aliran data untuk kelengkapan dan keakuratan
2. Berikan titik awal untuk mengembangkan layar dan laporan
3. Tentukan isi data yang disimpan dalam file
4. Kembangkan logika untuk proses diagram aliran data
5. Membuat *XML* (bahasa *markup* yang dapat diperluas)

Kamus data dibuat dengan cara memeriksa dan mendeskripsikan isi dari arus data, *datastore*, dan proses. Setiap penyimpanan data dan aliran data harus didefinisikan dan kemudian diperluas untuk memasukkan rincian elemen yang dikandungnya. Logika dari setiap proses harus dijelaskan menggunakan data yang mengalir masuk atau keluar dari proses. Kelalaian dan kesalahan desainer lainnya harus dicatat dan diselesaikan [8].

Struktur data biasanya dijelaskan menggunakan notasi aljabar. Metode ini memungkinkan penganalisis untuk menghasilkan tampilan dari elemen yang menyusun struktur data bersama dengan informasi tentang elemen tersebut. Misalnya, penganalisis akan menunjukkan apakah ada banyak elemen yang sama dalam struktur data (grup berulang), atau apakah dua elemen mungkin ada secara eksklusif satu sama lain. Notasi aljabar menggunakan simbol-simbol berikut [8]:

1. Tanda sama dengan = berarti "terdiri dari".
2. Tanda plus + berarti "dan".
3. *Braces* {} menunjukkan elemen berulang, juga disebut grup atau tabel berulang. Mungkin ada satu elemen berulang atau beberapa dalam grup. Grup berulang mungkin memiliki kondisi, seperti jumlah pengulangan yang tetap, atau batas atas dan bawah untuk jumlah pengulangan.
4. *Braket* [] mewakili situasi salah satu / atau. Salah satu elemen mungkin ada atau yang lain, tetapi tidak keduanya. Elemen-elemen yang tercantum di antara tanda kurung saling eksklusif.

5. Tanda kurung ( ) mewakili elemen opsional. Elemen opsional dapat dikosongkan pada layar entri dan mungkin berisi spasi atau nol untuk bidang numerik dalam struktur file.

Berikut ini adalah contoh kamus data [8]:



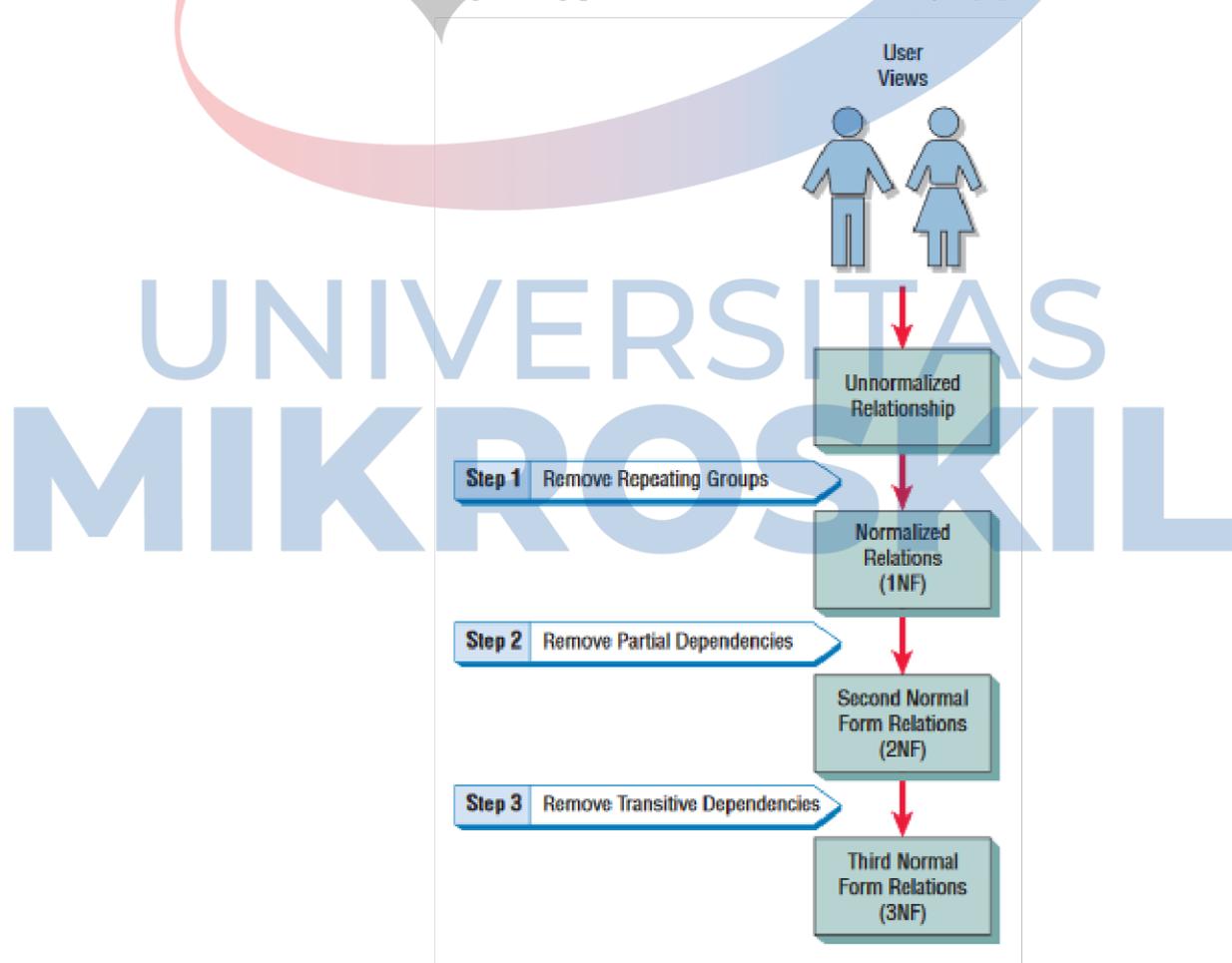
Gambar 2. 6 Contoh Kamus Data

#### 2.5.4 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 dinormalisasi lebih mudah dipelihara daripada struktur data lainnya [8].

Setiap langkah melibatkan prosedur penting, yang menyederhanakan struktur data. Relasi yang berasal dari tampilan pengguna atau penyimpanan data kemungkinan besar tidak akan dinormalisasi. Tahap pertama dari proses ini termasuk menghapus semua grup yang berulang dan mengidentifikasi kunci primer. Untuk melakukannya, relasi perlu dipecah menjadi dua atau lebih relasi. Pada titik ini, relasi mungkin sudah menjadi bentuk normal ketiga, tetapi kemungkinan lebih banyak langkah yang diperlukan untuk mengubah relasi ke bentuk normal ketiga. Langkah kedua memastikan bahwa semua atribut nonkunci sepenuhnya bergantung pada kunci utama. Semua dependensi parsial dihapus dan ditempatkan di relasi lain. Langkah ketiga menghapus dependensi transitif. Ketergantungan transitif adalah ketergantungan di mana atribut nonkunci bergantung pada atribut nonkunci lainnya [8].



Gambar 2. 7 Tahapan Normalisasi

Berikut ini adalah contoh normalisasi dari suatu tabel [8]:

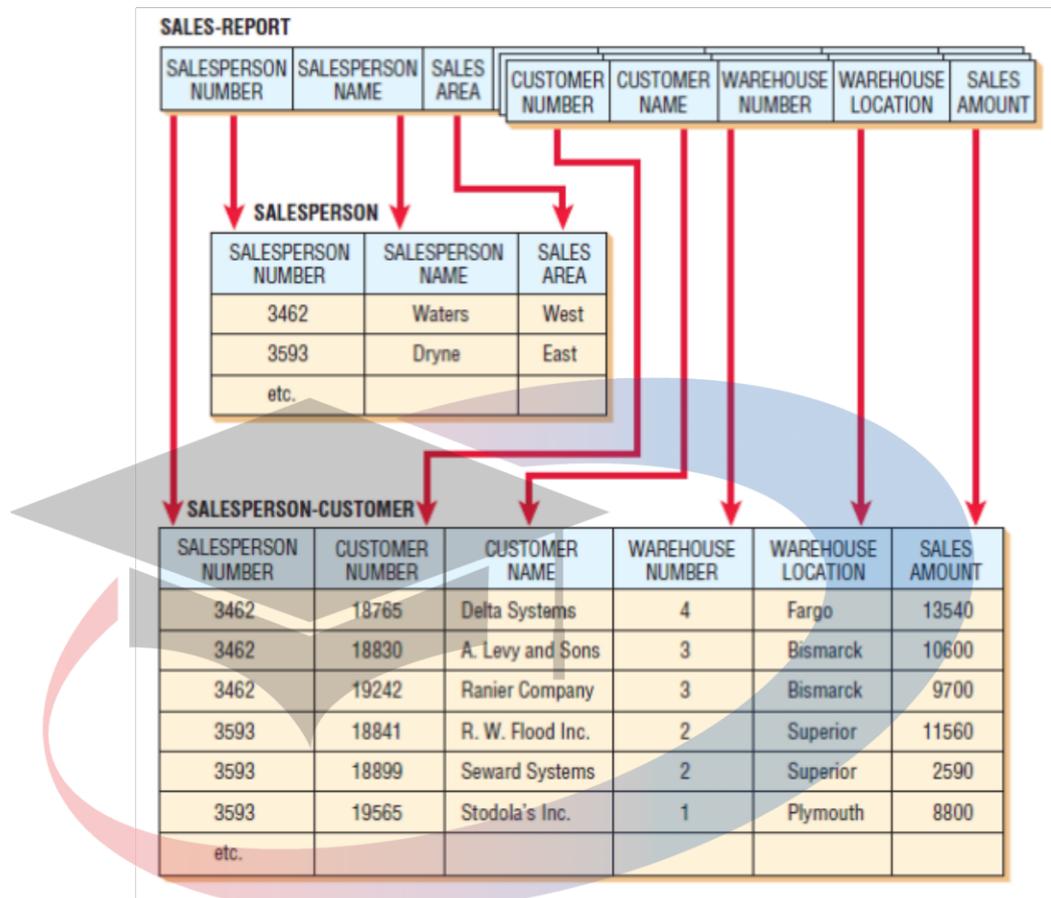
SALESPERSON NUMBER	SALESPERSON NAME	SALES AREA	CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES AMOUNT
3462	Waters	West	18765	Delta Systems	4	Fargo	13540
			18830	A. Levy and Sons	3	Bismarck	10600
			19242	Ranier Company	3	Bismarck	9700
3593	Dryne	East	18841	R. W. Flood Inc.	2	Superior	11560
			18899	Seward Systems	2	Superior	2590
			19565	Stodola's Inc.	1	Plymouth	8800
etc.							

Gambar 2. 8 Tabel yang Belum dinormalisasikan

Karena relasi ini didasarkan pada tampilan pengguna awal, maka disebut sebagai *SALES-REPORT*. *SALES-REPORT* adalah relasi yang tidak dinormalisasi, karena memiliki grup yang berulang. Penting juga untuk diperhatikan bahwa atribut tunggal seperti *SALESPERSON-NUMBER* tidak dapat berfungsi sebagai kunci [8].

1. Bentuk Normalisasi Pertama (*1NF*)

# UNIVERSITAS MIKROSKIL



Gambar 2. 9 Bentuk Normalisasi Pertama (1NF)

Langkah pertama dalam menormalkan relasi adalah menghapus grup berulang. Dalam contoh, relasi *SALES-REPORT* yang tidak dinormalisasi akan dipecah menjadi dua relasi yang terpisah. Relasi baru ini akan diberi nama *SALESPERSON* dan *SALESPERSON-CUSTOMER* [8].

Gambar 2.6 menunjukkan bagaimana relasi asli yang tidak dinormalisasi *SALES-REPORT* dinormalisasi dengan memisahkan relasi menjadi dua relasi baru. Perhatikan bahwa hubungan *SALESPERSON* berisi kunci utama *SALESPERSON-NUMBER* dan semua atribut yang tidak berulang (*SALESPERSON-NAME* dan *SALES-AREA*) [8].

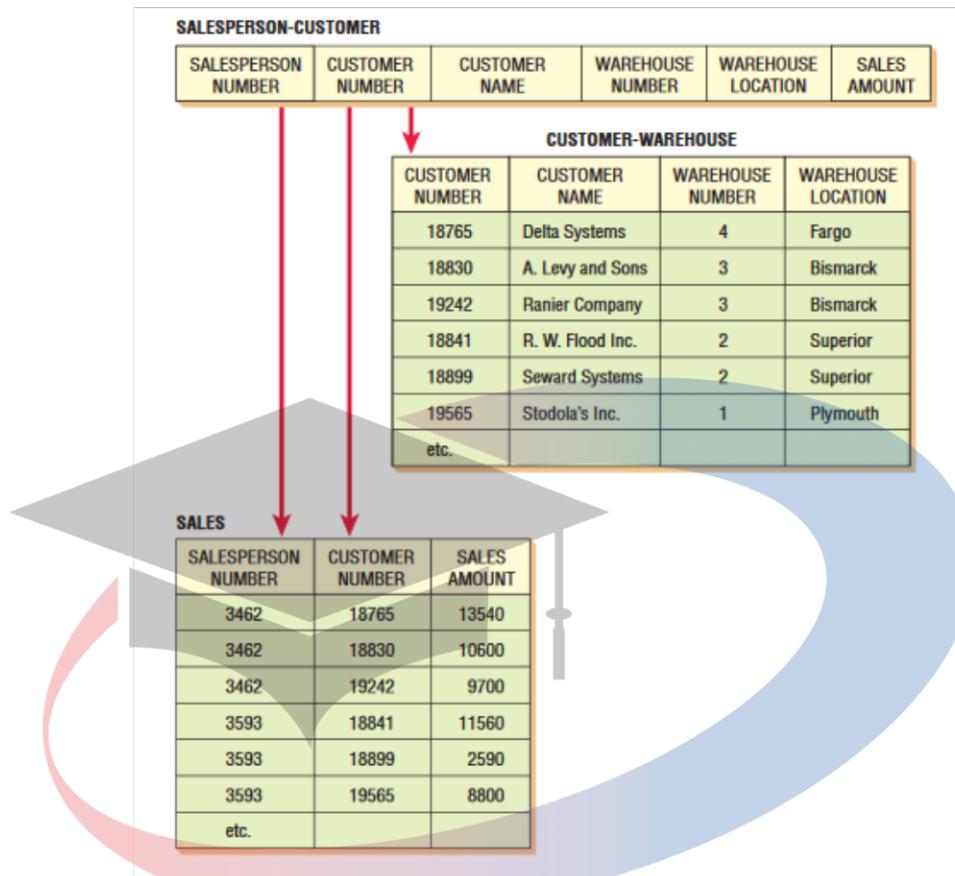
Relasi kedua, *SALESPERSON-CUSTOMER*, berisi kunci utama dari relation *SALESPERSON* (kunci utama *SALESPERSON* adalah *SALESPERSON-NUMBER*), serta semua atribut yang merupakan bagian dari grup berulang (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*, *WAREHOUSE-NUMBER*, *WAREHOUSE-*

*LOCATION*, dan *SALES-AMOUNT*). Mengetahui *SALESPERSON-NUMBER*, bagaimanapun, tidak berarti bahwa pengguna secara otomatis akan mengetahui *CUSTOMER-NAME*, *SALES-AMOUNT*, *WAREHOUSE-LOCATION*. Dalam hubungan ini, seseorang harus menggunakan kunci yang digabungkan (baik *SALESPERSON-NUMBER* dan *CUSTOMER-NUMBER*) untuk mengakses informasi lainnya. Relasi dapat ditulis dalam notasi singkat sebagai berikut [8]:



Relasi *SALESPERSON-CUSTOMER* adalah relasi normal pertama, tetapi tidak dalam bentuk idealnya. Masalah muncul karena beberapa atribut tidak bergantung secara fungsional pada kunci primer. Relasi asli yang tidak dinormalisasi *SALES-REPORT* dipisahkan menjadi dua relasi, *SALESPERSON* (3NF) dan *SALESPERSON-CUSTOMER* (1NF) [8].

## 2. Bentuk Normalisasi Kedua (2NF)



Gambar 2. 10 Normalisasi Kedua (2NF)

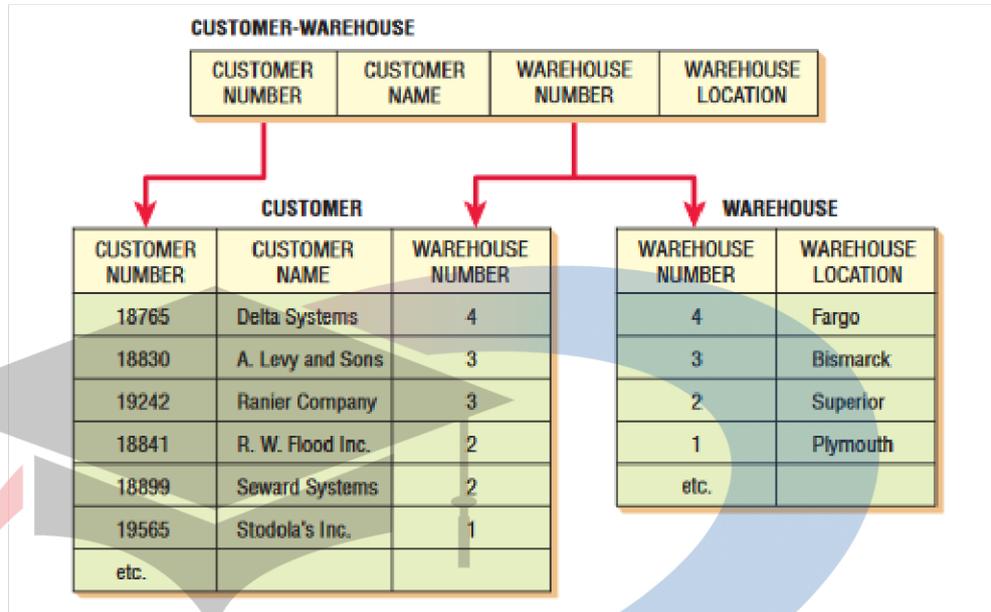
Dalam bentuk normal kedua, semua atribut akan secara fungsional bergantung pada kunci utama. Oleh karena itu, langkah selanjutnya adalah menghapus semua atribut yang sebagian bergantung dan menempatkannya di relasi lain [8]. Gambar 2.7 menunjukkan bagaimana relasi *SALESPERSON-CUSTOMER* dibagi menjadi dua relasi baru *SALES* dan *CUSTOMER-WAREHOUSE*. Hubungan juga dapat diekspresikan sebagai berikut [8]:

*SALES* (*SALESPERSON-NUMBER*,  
*CUSTOMER-NUMBER*,  
*SALES-AMOUNT*)

and

*CUSTOMER WAREHOUSE* (*CUSTOMER-NUMBER*,  
*CUSTOMER-NAME*,  
*WAREHOUSE-NUMBER*,  
*WAREHOUSE-LOCATION*)

### 3. Bentuk Normalisasi Ketiga (3NF)



Gambar 2. 11 Normalisasi Ketiga (3NF)

Relasi yang dinormalisasi ada dalam bentuk normal ketiga jika semua atribut nonkunci secara fungsional bergantung sepenuhnya pada kunci primer dan tidak ada ketergantungan transitif (nonkunci) [8]. Dengan cara yang mirip dengan langkah sebelumnya, adalah mungkin untuk memisahkan relasi *CUSTOMER-WAREHOUSE* menjadi dua relasi, seperti yang ditunjukkan pada Gambar 2.8.

Kedua relasi baru tersebut disebut *CUSTOMER* dan *WAREHOUSE*, dan dapat ditulis sebagai berikut [8]:

*CUSTOMER* (*CUSTOMER-NUMBER*,  
*CUSTOMER-NAME*,  
*WAREHOUSE-NUMBER*)

and

*WAREHOUSE* (*WAREHOUSE-NUMBER*,  
*WAREHOUSE-LOCATION*)

Kunci utama untuk relasi *CUSTOMER* adalah *CUSTOMER-NUMBER*, dan kunci utama untuk relasi *WAREHOUSE* *WAREHOUSE-NUMBER* [8].

#### 2.5.5 Basis Data

Basis data bukan hanya kumpulan file. Sebaliknya, basis data adalah sumber data pusat yang dimaksudkan untuk dibagikan oleh banyak pengguna untuk berbagai aplikasi. Inti dari basis data adalah *data base management system (DBMS)*, yang memungkinkan pembuatan, modifikasi, pembaruan basis data, pengambilan data, pembuatan laporan dan tampilan. Orang yang memastikan bahwa basis data memenuhi tujuannya disebut *administrator database* [8].

Tujuan efektivitas basis data meliputi [8]:

1. Memastikan bahwa data dapat dibagikan di antara pengguna untuk berbagai aplikasi.
2. Menjaga data yang akurat dan konsisten.
3. Memastikan bahwa semua data yang diperlukan untuk aplikasi saat ini dan di masa mendatang akan tersedia.
4. Mengizinkan basis data berkembang seiring kebutuhan pengguna tumbuh.
5. Mengizinkan pengguna untuk membuat tampilan pengguna atas data tanpa memperhatikan cara data disimpan secara fisik.

Daftar tujuan di atas memberi peringatan tentang keuntungan dan kerugian dari pendekatan basis data. Pertama, berbagi data berarti bahwa data hanya perlu disimpan satu kali, yang pada gilirannya membantu mencapai integritas data, karena perubahan pada data dilakukan dengan lebih mudah jika data muncul sekali daripada di banyak file yang berbeda [8].

Ketika seorang pengguna membutuhkan data tertentu, basis data yang dirancang dengan baik mengantisipasi kebutuhan akan data tersebut (atau mungkin sudah digunakan untuk aplikasi lain). Akibatnya, data memiliki probabilitas yang lebih tinggi untuk tersedia dalam basis data dibandingkan dengan sistem file konvensional. Basis data yang dirancang dengan baik juga bisa lebih fleksibel daripada file terpisah. Artinya basis data dapat berkembang seiring dengan perubahan kebutuhan pengguna dan aplikasi [8].