

BAB II TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. [4]

Sistem yang baik memiliki karakteristik sebagai berikut [4]:

1. Komponen

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber data

mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan sistem (*input*)

Masukan adalah energi yang dimasukkan ke dalam sistem yang dapat berupa perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer, program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh, komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah sistem

Suatu sistem menjadi bagian pengolah yang akan mengubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan *input* yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.1.2 Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya [4]. Sumber informasi adalah data [4]. Data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol, gambar, angka, huruf, atau simbol-simbol yang menunjukkan suatu ide, objek, kondisi atau situasi dan lain - lain [5].

Data menggambarkan suatu kejadian yang sedang terjadi, dimana data tersebut akan diolah dan diterapkan dalam sistem menjadi input yang berguna dalam suatu sistem. Kualitas informasi tergantung pada beberapa hal yaitu [6]:

1. Relevan (*relevancy*)

Informasi yang berkualitas akan mampu menunjukkan benang merah relevansi kejadian masa lalu, hari ini, dan masa depan sebagai sebuah bentuk aktivitas yang kongkrit dan mampu dilaksanakan dan dibuktikan oleh siapa saja.

2. Akurat (*accuracy*)

Suatu informasi dikatakan berkualitas, jika seluruh kebutuhan informasi tersebut telah tersampaikan, seluruh pesan telah benar/sesuai, serta pesan yang disampaikan sudah lengkap.

3. Tepat waktu (*timeliness*)

Berbagai proses dapat diselesaikan dengan tepat waktu, laporan-laporan yang dibutuhkan dapat disampaikan dengan tepat waktu.

4. Ekonomis (*economy*)

Informasi yang dihasilkan mempunyai daya jual yang tinggi, serta biaya operasional untuk menghasilkan informasi tersebut minimal.

5. Efisiensi (*efficiency*)

Informasi yang berkualitas memiliki sintaks ataupun kalimat yang sederhana, namun mampu memberikan makna dan hasil yang mendalam.

6. Dapat dipercaya (*reliability*)

Informasi tersebut berasal dari sumber yang dapat dipercaya. Sumber tersebut juga telah teruji tingkat kejujurannya.

Dapat disimpulkan, bahwa informasi merupakan hasil dari pengolahan data yang diklasifikasikan atau diolah untuk digunakan dalam proses pengambilan keputusan sehingga hasilnya dapat bermanfaat dalam operasional dan manajemen.

2.1.3 Sistem Informasi

Sistem informasi merupakan suatu kombinasi teratur apapun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi [7].

Fungsi Sistem Informasi [7]:

1. Untuk meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan perantara sistem informasi.
2. Memperbaiki Produktivitas aplikasi pengembangan dan pemeliharaan sistem.

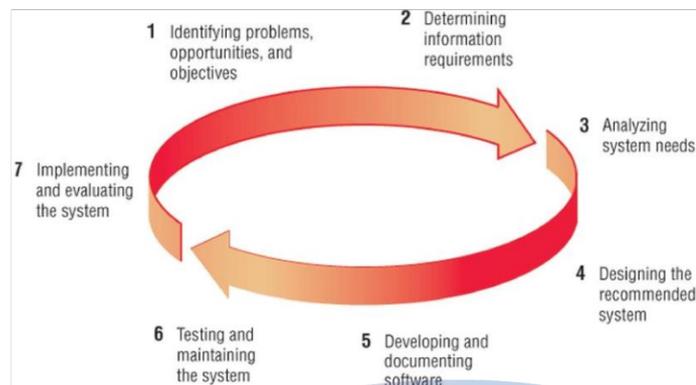
3. Menjamin tersedianya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
4. Mengidentifikasi kebutuhan mengenai keterampilan pendukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.
6. Menetapkan investasi yang akan diarahkan pada sistem informasi.
7. Mengembangkan proses perencanaan yang efektif.

Komponen-komponen dari sistem informasi adalah sebagai berikut [7]:

1. Komponen *input*, adalah data yang masuk ke dalam sistem informasi.
2. Komponen model, adalah kombinasi prosedur, logika, dan model matematika yang memroses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *output*, adalah hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen teknologi, adalah alat dalam sistem informasi yang digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output*, serta memantau pengendalian sistem.
5. Komponen basis data, adalah kumpulan data yang saling berhubungan yang tersimpan di dalam komputer dengan menggunakan *software database*.
6. Komponen kontrol, adalah komponen yang mengendalikan gangguan terhadap sistem informasi.

2.2 Siklus Hidup Pengembangan Sistem

System 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 ditunjukkan pada gambar berikut ini [3].



Gambar 2.1 Siklus Hidup Pengembangan Sistem

Tahapan-tahapan siklus hidup pengembangan sistem terdiri dari [3]:

1. Mengidentifikasi masalah, peluang, dan tujuan

Pada tahap pertama dari siklus hidup pengembangan sistem ini, penganalisis mengidentifikasi masalah, peluang, dan tujuan-tujuan yang hendak dicapai.

a. Mengidentifikasi masalah

Tahap ini berarti bahwa penganalisis melihat dengan jujur apa yang terjadi di bisnis, kemudian bersama-sama dengan anggota organisasi lainnya, penganalisis menentukan dengan tepat masalah-masalah yang dihadapi oleh organisasi.

b. Peluang

Merupakan situasi dimana penganalisis yakin bahwa peningkatan dapat dilakukan melalui penggunaan sistem informasi terkomputerisasi. Mengukur peluang memungkinkan bisnis untuk mencapai sisi kompetitif atau menyusun standar-standar industri.

c. Tujuan

Mengidentifikasi tujuan juga merupakan komponen terpenting dalam tahap pertama ini. Penganalisis harus menemukan apa yang sedang dilakukan dalam bisnis sehingga penganalisis bisa memastikan bahwa penggunaan sistem informasi akan membantu bisnis dalam mencapai tujuan-tujuannya.

2. Menentukan syarat - syarat informasi

Di tahap ini penganalisis menentukan syarat - syarat informasi di dalam bisnis dengan cara menentukan sampel dan memeriksa data mentah, wawancara, mengamati perilaku keputusan dan lingkungan kantor, serta *prototyping*.

Penganalisis berusaha untuk memahami informasi apa yang dibutuhkan pemakai agar bisa ditampilkan dalam pekerjaan mereka dan bagaimana caranya agar sistem usulan benar-benar bermanfaat bagi penggunanya. Orang-orang yang terlibat dalam tahap ini adalah penganalisis dan pemakai, juga manajer operasi dan pegawai operasional. Penganalisis sistem perlu mengetahui detail-detail fungsi sistem berjalan: siapa (orang yang terlibat), apa (kegiatan bisnis), di mana (lingkungan di mana pekerjaan itu dilakukan), kapan (waktu yang tepat), dan bagaimana (bagaimana prosedur yang harus dijalankan) dari bisnis yang sedang dipelajari.

3. Menganalisis kebutuhan sistem

Di tahap ini perangkat dan teknik-teknik tertentu akan membantu penganalisis menentukan kebutuhan. Perangkat dan teknik-teknik yang dimaksud ialah penggunaan diagram aliran data untuk menyusun daftar *input*, *proses*, dan *output* fungsi bisnis dalam bentuk grafik terstruktur. Dari diagram aliran data dikembangkan suatu kamus data yang berisikan daftar seluruh item data yang digunakan dalam sistem, termasuk juga spesifikasinya. Penganalisis juga menganalisis keputusan terstruktur yang dibuat. Keputusan terstruktur adalah keputusan-keputusan dimana kondisi, kondisi alternatif, tindakan, serta aturan tindakan ditetapkan. Ada tiga metode utama untuk menganalisis keputusan terstruktur, yaitu bahasa Inggris terstruktur, rancangan keputusan, dan pohon keputusan.

4. Merancang sistem yang direkomendasikan

Di tahap ini penganalisis sistem menggunakan informasi-informasi yang terkumpul sebelumnya untuk membuat rancangan logis dan fisik sistem usulan. Penganalisis merancang prosedur sedemikian rupa sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Selain itu, penganalisis menggunakan teknik perancangan antarmuka yang baik untuk menjamin keefektifan *input* sistem informasi. Rancangan logis dan fisik sistem usulan juga berkaitan dengan interaksi manusia dan komputer. Antarmuka menghubungkan pemakai dengan sistem, jadi perannya benar-benar sangat penting. Peran pemakai dalam perancangan antarmuka sangat penting agar sistem yang dihasilkan mudah dimengerti, aman, menarik, dan mudah dipakai. Contoh dari antarmuka pemakai

adalah *keyboard* (untuk mengetik pertanyaan dan jawaban), menu-menu pada layar (untuk mendapatkan perintah pemakai), serta berbagai jenis *Graphical User Interface* (GUI) yang menggunakan tetikus atau layar sentuh. Tahap perancangan juga mencakup perancangan basis data yang bisa menyimpan data yang diperlukan oleh pembuat keputusan. Pemakai akan diuntungkan oleh basis data yang tersusun dengan baik sehingga mudah dimengerti dan sesuai dengan cara pemakai bekerja. Pada tahap ini, penganalisis bekerja sama dengan pemakai untuk merancang *output* (baik pada layar maupun hasil cetakan) yang dapat memenuhi kebutuhan pemakai. Terakhir, penganalisis harus merancang prosedur-prosedur *back up* dan kontrol untuk melindungi sistem dan data serta untuk membuat paket-paket spesifikasi program bagi pemrogram. Setiap paket bisa terdiri dari *layout input* dan *ouput*, spesifikasi *file* dan detail-detail proses, pohon keputusan atau tabel keputusan, diagram aliran data, serta nama-nama dan fungsi-fungsi subprogram yang sudah tertulis.

5. Mengembangkan dan mendokumentasikan perangkat lunak

Di tahap ini penganalisis bekerja bersama-sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang diperlukan. Penganalisis juga bekerja sama dengan pemakai untuk mengembangkan dokumentasi perangkat lunak yang efektif, mencakup melakukan prosedur secara manual, bantuan *online*, dan *website*-nya yang membuat fitur *Frequently Asked Questions*. Kegiatan dokumentasi menunjukkan kepada pemakai tentang cara penggunaan perangkat lunak dan apa yang harus dilakukan jika perangkat lunak mengalami masalah. Pemrogram adalah pelaku utama dalam tahap ini karena mereka merancang, membuat kode, dan mengatasi kesalahan-kesalahan dari program komputer. Untuk memastikan kualitasnya, pemrogram bisa membuat perancangan dan kode program yang akan dijalankan, serta menjelaskan bagian-bagian kompleks dari program kepada tim pemrogram lainnya.

6. Menguji dan mempertahankan sistem

Sebelum sistem informasi dapat digunakan, maka terlebih dahulu harus dilakukan pengujian. Akan bisa menghemat biaya bila dapat menangkap adanya masalah sebelum sistem tersebut ditetapkan. Sebagian pengujian dilakukan oleh pemrogram sendiri, dan lainnya dilakukan oleh penganalisis sistem. Rangkaian

pengujian ini pertama-tama dijalankan bersama-sama dengan data contoh serta dengan data aktual dari sistem yang telah ada. Mempertahankan sistem dan dokumentasinya dimulai di tahap ini dan dilakukan secara rutin selama sistem informasi dijalankan. Sebagian besar kerja rutin pemrogram adalah melakukan pemeliharaan dan bisnis menghabiskan banyak uang untuk kegiatan pemeliharaan. Kegiatan pemeliharaan seperti memperbarui program bisa dilakukan secara otomatis melalui *website vendor*. Sebagian besar prosedur sistematis yang dijalankan penganalisis selama siklus hidup pengembangan sistem membantu memastikan bahwa pemeliharaan bisa dijaga sampai tingkat minimum.

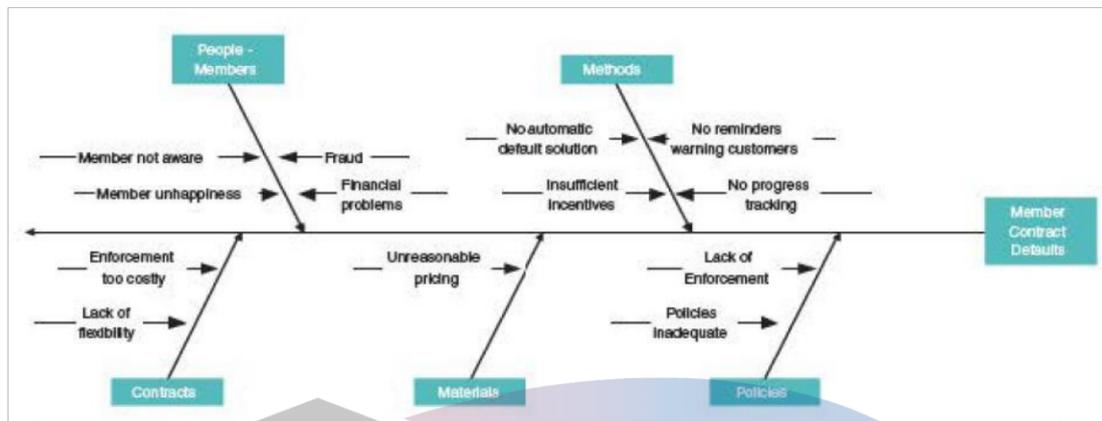
7. Mengimplementasikan dan mengevaluasi sistem

Penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai dalam menggunakan sistem. Sebagian pelatihan tersebut dilakukan oleh *vendor*, namun kesalahan pelatihan merupakan tanggung jawab penganalisis sistem. Selain itu, penganalisis perlu merencanakan konversi perlahan dari sistem lama ke sistem baru. Proses ini mencakup perubahan *file* dari format lama ke format baru, atau membangun suatu basis data, menginstalasi peralatan, dan membawa sistem baru untuk diproduksi. Evaluasi yang ditunjukkan sebagai bagian dari tahap terakhir dari SDLC biasanya dimaksudkan untuk pembahasan. Sebenarnya evaluasi dilakukan di setiap tahap. Kriteria utama yang harus dipenuhi adalah apakah pemakai yang dituju benar-benar menggunakan sistem.

2.3 Teknik Pengembangan Sistem

2.3.1 Diagram Ishikawa (*Fishbone Diagram*)

Diagram Ishikawa adalah sebuah tool grafis yang digunakan untuk mengidentifikasi, mengeksplorasi, dan menggambarkan masalah dan sebabnya serta dampak dari masalah tersebut. Diagram Ishikawa disebut juga sebagai Diagram Sebab-dan-Akibat atau Diagram *Fishbone*, karena kemiripan akan diagram ini dengan bentuk seekor ikan [8].



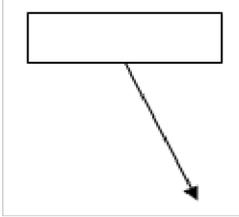
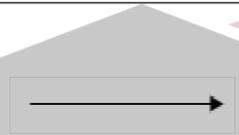
Gambar 2.2 Contoh Diagram Ishikawa

Diagram *fishbone* pertama kali dicetuskan oleh Kaoru Ishikawa, yang membuat terobosan mengenai proses manajemen kualitas di galangan kapal Kawasaki, Jepang dan dalam prosesnya, Ishikawa menjadi salah satu pencetus manajemen modern [8].

Penggambaran diagram *fishbone* dimulai dengan peletakan nama masalah yang ingin digambarkan pada sebelah kanan diagram (kepala ikan). Kemudian, sebab dari masalah tersebut digambarkan sebagai tulang ikan di sepanjang tulang belakang, dengan setiap sebab masalah digambarkan dalam sebuah anak panah yang menunjuk ke arah tulang belakang. Biasanya “tulang” ini dilabeli empat kategori utama yaitu *materials* (barang), *machines* (mesin), *manpower* (tenaga manusia), dan *methods* (metode), yang disebut sebagai *the four Ms*. Nama lainnya dapat disesuaikan dengan masalah yang terjadi. Kategori alternatif atau tambahan meliputi *places* (tempat), *procedures* (prosedur), *policies* (kebijakan), dan *people* (manusia), yang disebut sebagai *the four Ps* atau *surrounding* (suasana), *suppliers* (*supplier*), *systems* (sistem), dan *skills* (kemampuan), yang disebut sebagai *the four Ss* [8].

Tabel 2.1 Tabel Notasi Diagram *Fishbone*

| No | Notasi | Keterangan |
|----|--------|---|
| 1 | | <i>Effect</i> , merupakan simbol yang menunjukkan akibat dari masalah yang digambarkan pada diagram <i>fishbone</i> dan |

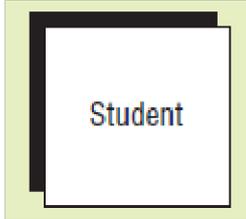
| | | |
|---|---|---|
| | | akibat tersebut digambarkan di sebelah kanan anak panah. |
| 2 |  | <i>Category</i> , menunjukkan kategori dari masalah yang digambarkan pada diagram <i>fishbone</i> , dimana anak panah simbol ini menunjuk langsung ke anak panah yang mengarah ke <i>effect</i> . |
| 3 |  | <i>Cause</i> , merupakan sebab dari masalah yang digambarkan pada diagram <i>fishbone</i> , dimana anak panah simbol ini menunjuk anak panah simbol <i>category</i> . |

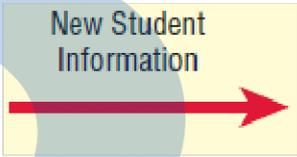
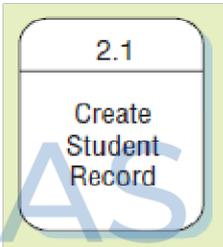
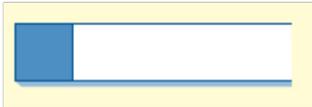
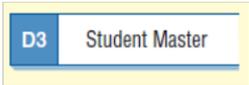
2.3.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah menggambarkan pandangan mengenai masukan, proses, dan keluaran sistem. Penganalisis sistem berupaya memahami syarat - syarat informasi pengguna, yaitu harus mampu mengkonseptualisasikan dimana data melalui dan apa keluarannya [3].

Dengan menggunakan kombinasi dari empat simbol, penganalisis sistem dapat menciptakan suatu gambaran proses yang bisa menampilkan dokumentasi sistem yang solid. Empat simbol dasar yang digunakan untuk memetakan gerakan DFD adalah [3]:

Tabel 2.2 Tabel Simbol Dasar dalam DFD

| Gambar | Arti | Fungsi | Contoh |
|---|---------|--|---|
|  | Entitas | Untuk menggambarkan suatu entitas eksternal (bagian lain, sebuah perusahaan, seseorang, atau sebuah mesin) |  |

| | | | |
|---|-------------------|---|---|
| | | yang dapat mengirim data atau menerima data dari sistem | |
|  | <i>Data Flow</i> | Untuk menunjukkan perpindahan data dari satu titik ke titik yang lain, dengan kepala tanda panah mengarah ke tujuan data. |  |
|  | Proses | Untuk menunjukkan adanya proses transformasi. |  |
|  | <i>Data Store</i> | Untuk menunjukkan penyimpanan data yang memungkinkan penambahan dan perolehan data. |  |

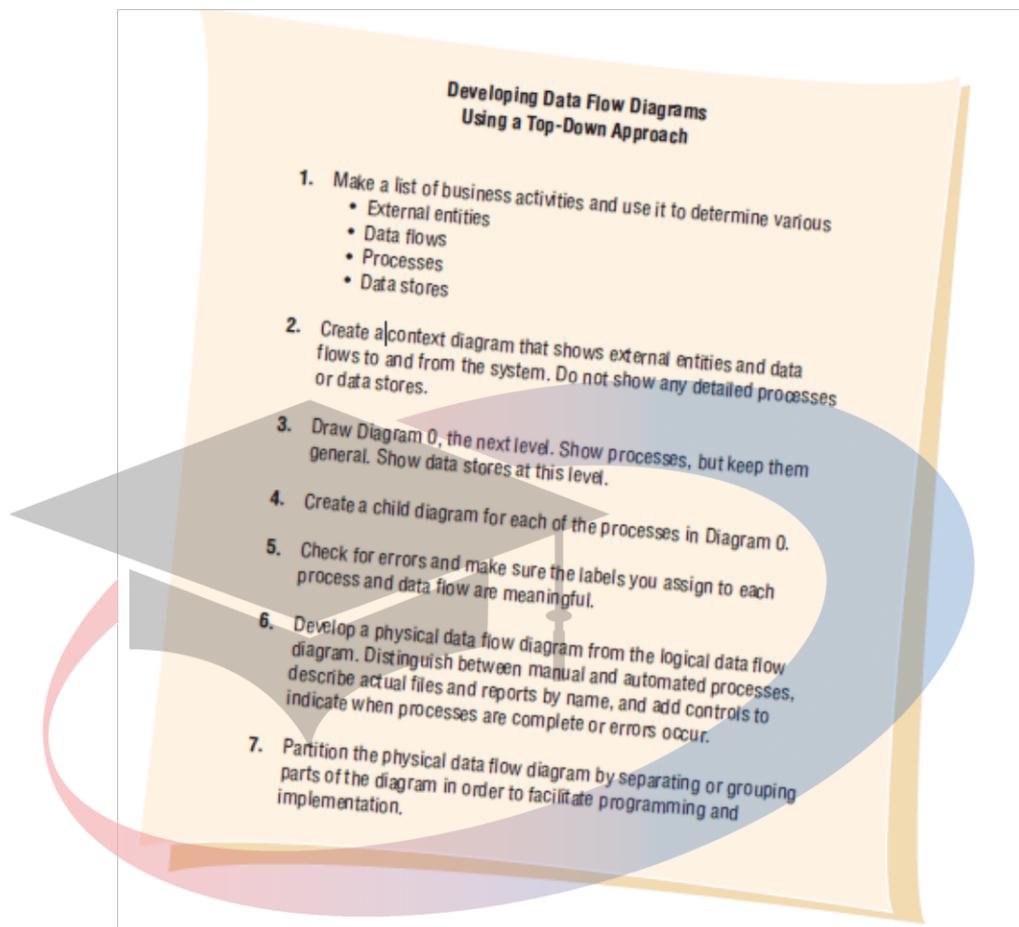
Beberapa kesalahan umum yang dibuat saat menggambar diagram aliran data adalah sebagai berikut [3]:

1. Lupa memasukkan suatu aliran data atau mengarahkan kepala anak panah pada arah yang salah. Contohnya adalah sebuah proses gambaran yang menunjukkan semua aliran data sebagai masukan atau sebagai keluaran saja.
2. Menghubungkan penyimpanan data dan entitas-entitas eksternal secara langsung satu sama lain. Penyimpanan data dan entitas eksternal hanya terhubung dengan suatu proses.
3. Aliran data atau proses-proses pemberian label yang tidak tepat.
4. Memasukkan lebih dari sembilan proses pada diagram aliran data sehingga menciptakan suatu diagram yang kacau yang akan memusingkan untuk dibaca dan malah menghalangi komunikasi.
5. Mengabaikan aliran data. Periksa aliran linear dalam diagram, artinya aliran data dimana setiap proses hanya memiliki satu masukan dan satu keluaran. Kecuali dalam hal diagram aliran data anak yang sangat mendetail, aliran data linear sangat jarang. Keberadaannya biasanya menunjukkan bahwa diagram tersebut kehilangan aliran data.
6. Menciptakan analisis yang tidak seimbang pada diagram anak. Masing-masing diagram anak harus memiliki masukan dan aliran data keluaran yang sama seperti proses induk.

Langkah-langkah menggambar diagram aliran data adalah sebagai berikut [3]:

1. Merancang Diagram Konteks (*Context Diagram*)

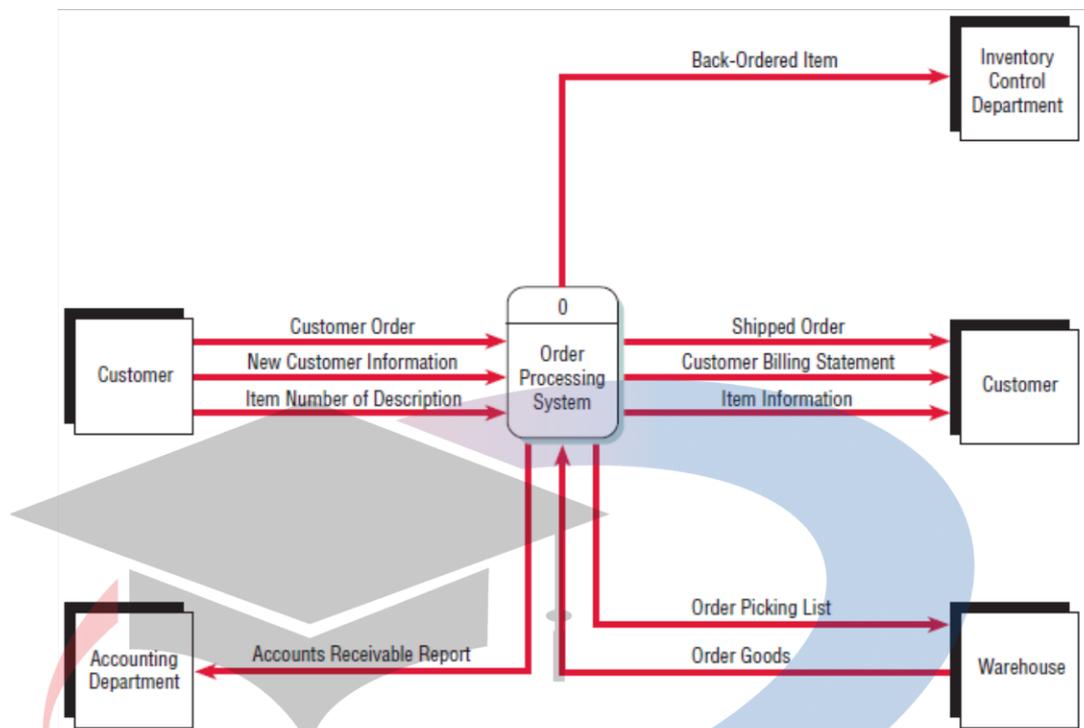
Dengan pendekatan atas-bawah untuk membuat diagram pengalihan data, diagram berganti dari umum ke khusus. Meskipun diagram pertama membantu penganalisis sistem memahami pengalihan data, sifat umumnya membatasi kegunaannya. Diagram konteks awal harus berupa suatu pandangan yang mencakup masukan-masukan dasar, sistem umum, dan keluaran. Diagram konteks adalah tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Langkah-langkah untuk mengembangkan diagram aliran data ditunjukkan pada gambar berikut ini.



Gambar 2 3 Langkah-Langkah untuk Mengembangkan DFD

Berikut ini adalah contoh penggambaran diagram konteks.

UNIVERSITAS
MIKROSKIL

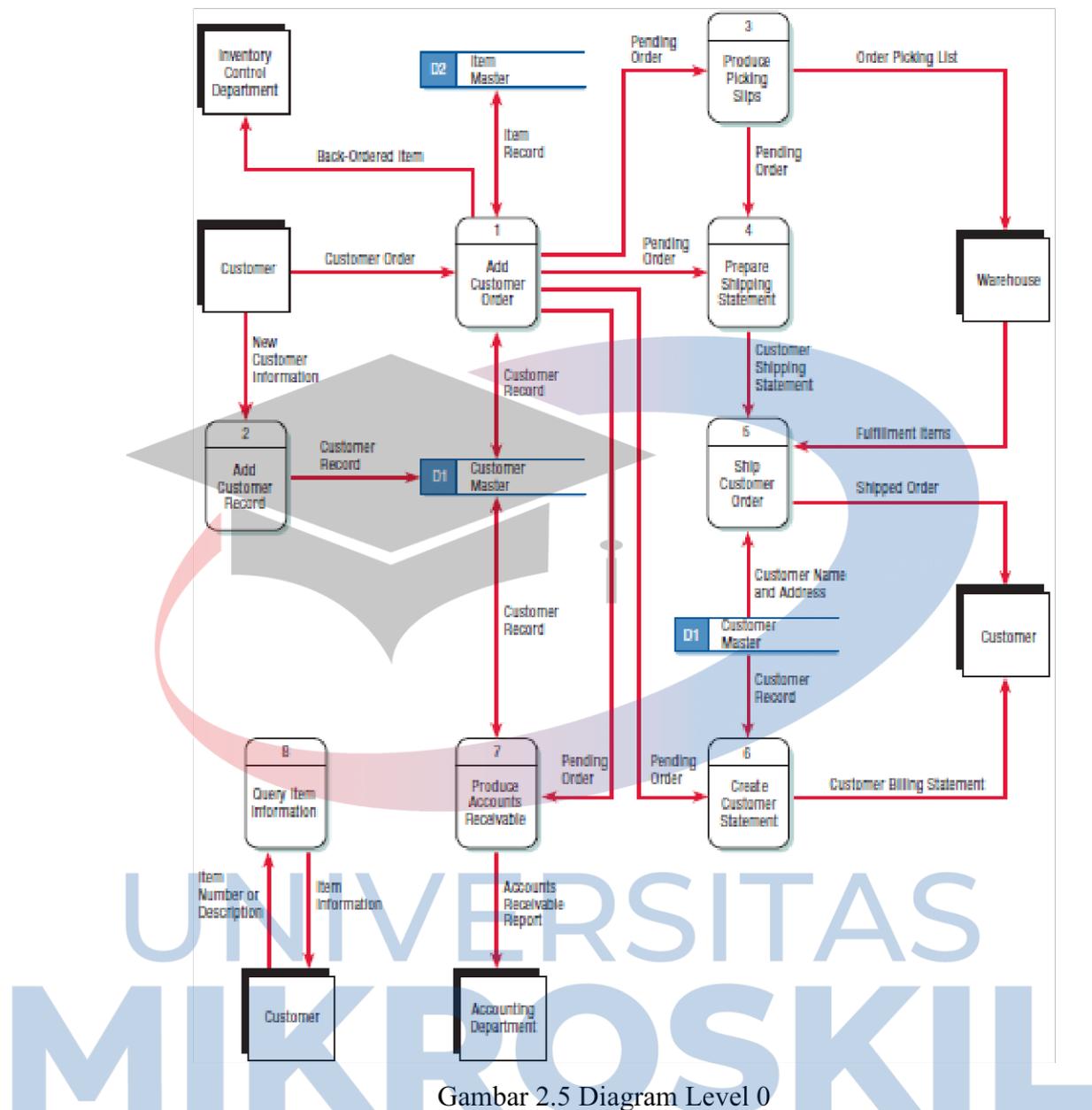


Gambar 2.4 Diagram Konteks

2. Merancang Diagram 0

Masukan dan keluaran yang ditetapkan dalam diagram konteks tetap konstan dalam semua diagram subrutannya. 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 nol adalah pengembangan diagram konteks dan bisa mencakup sampai sembilan proses. Memasukkan lebih banyak proses pada level ini akan membuat diagram menjadi kacau dan sulit dipahami. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Berikut ini adalah contoh penggambaran diagram level 0.

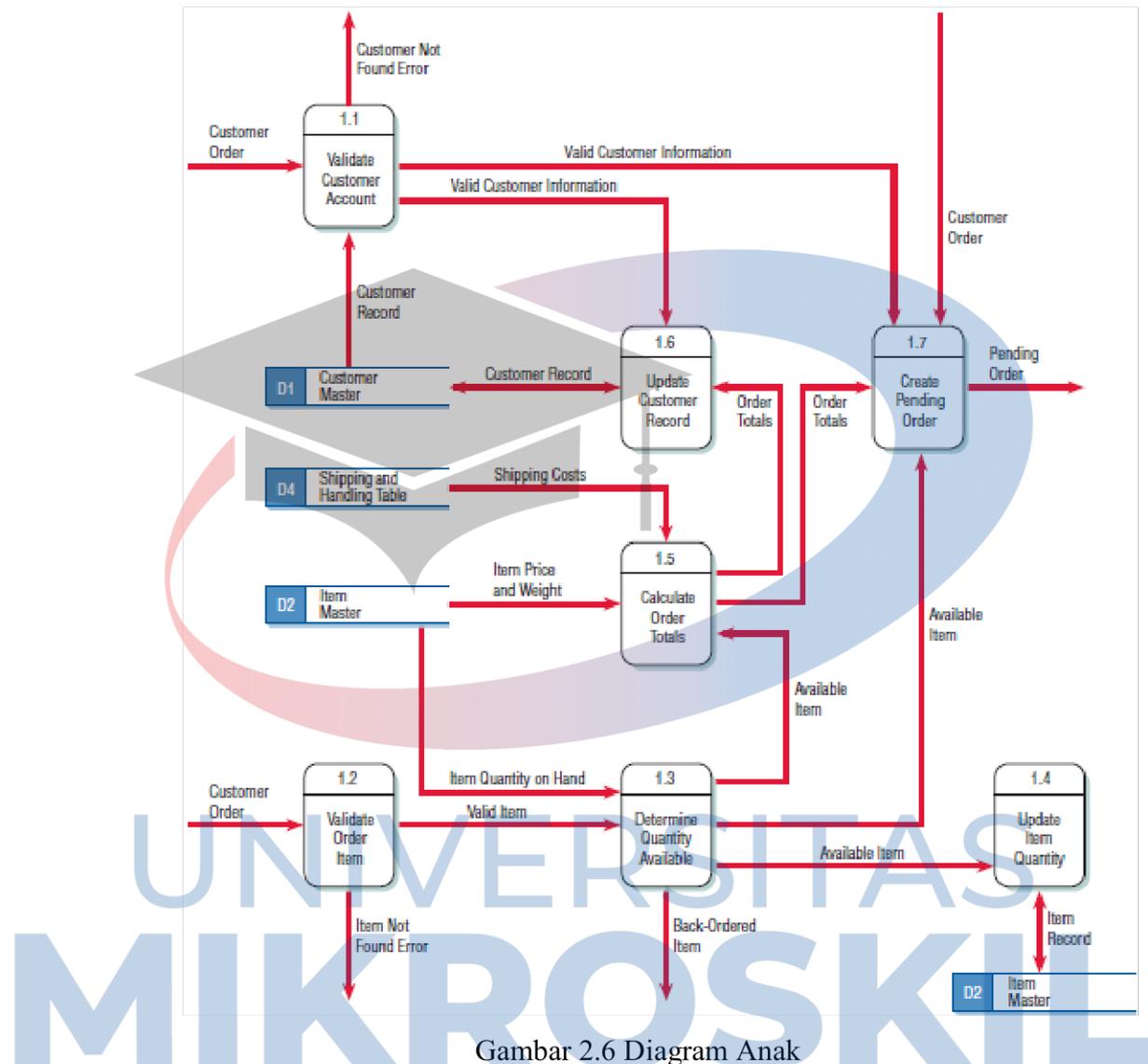


Gambar 2.5 Diagram Level 0

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

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 ke luar dari proses induk harus

ditunjukkan mengalir ke dalam atau ke luar dari diagram anak. Berikut ini adalah contoh penggambaran diagram anak.



Gambar 2.6 Diagram Anak

2.3.3 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 (maksudnya, *metadata*), suatu data yang disusun oleh penganalisis sistem untuk membimbing mereka selama melakukan analisis dan desain. Sebagai suatu dokumen, kamus data mengumpulkan dan mengkoordinasi istilah-istilah data tertentu, dan menjelaskan apa arti setiap istilah yang ada [3].

Sebagai tambahan untuk dokumentasi serta mengurangi redundansi, kamus data bisa digunakan untuk [3]:

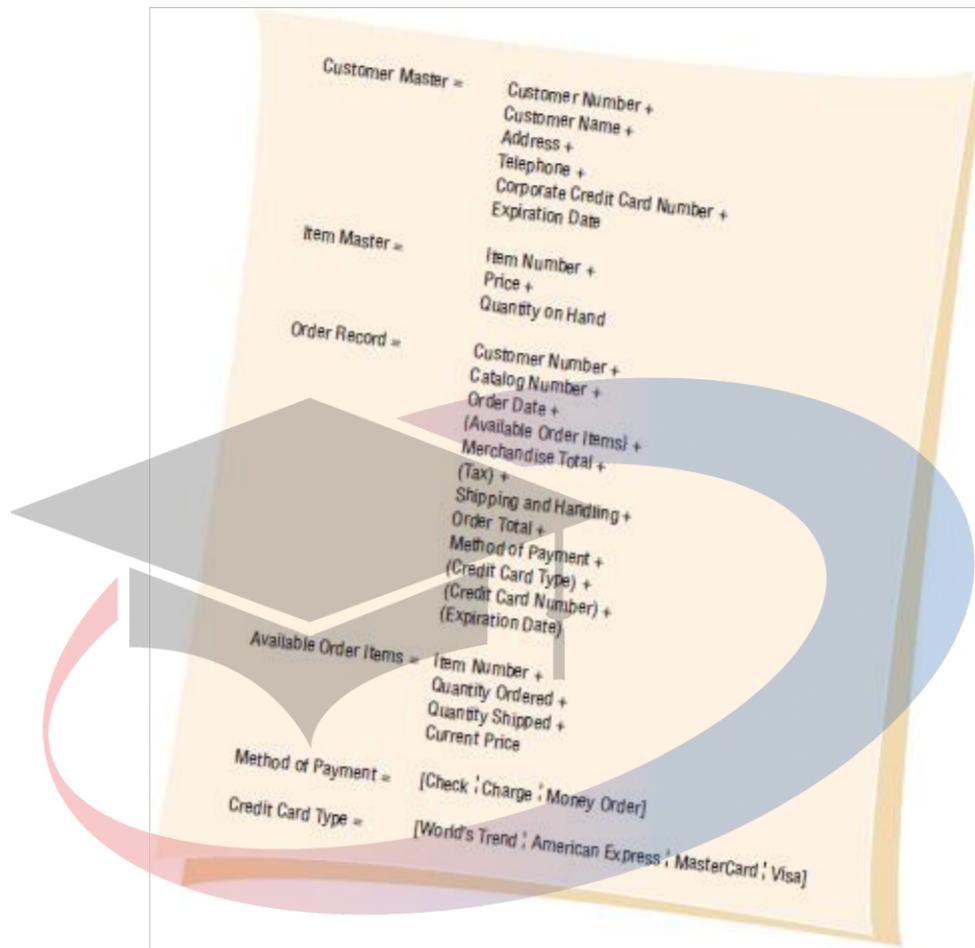
1. Menvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam *file-file*.
4. Mengembangkan logika untuk proses-proses diagram aliran data.
5. Membuat *Extensible Markup Language* (XML)

Kamus data dibuat dengan memperhatikan dan menggambarkan muatan aliran data, simpanan data, dan proses-proses. Setiap simpanan data dan aliran data harus ditetapkan dan kemudian diperluas sampai mencakup detail-detail elemen yang dimuatnya. Logika dari setiap proses ini bisa digambarkan dengan menggunakan data yang mengalir menuju dan ke luar dari proses tersebut. Ketidakhati-hatian dan kesalahan-kesalahan perancangan lain bisa ditegaskan dan dicari penyelesaiannya [3].

Struktur data biasanya digambarkan dengan menggunakan notasi aljabar. Metode ini memungkinkan penganalisis membuat suatu gambaran mengenai elemen-elemen yang membentuk struktur data bersama-sama dengan informasi-informasi mengenai elemen-elemen tersebut. Notasi aljabar menggunakan simbol-simbol sebagai berikut [3]:

1. Tanda sama dengan (=), artinya “terdiri dari”.
2. Tanda *plus* (+), artinya “dan”.
3. Tanda kurung {}, menunjukkan elemen-elemen repetitif, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang di dalam kelompok tersebut.
4. Tanda kurung [], menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada secara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain.
5. Tanda kurung (), menunjukkan suatu elemen yang bersifat pilihan. Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk *field-field* numerik pada struktur *file*.

Berikut ini adalah contoh pembuatan kamus data [3].



Gambar 2.7 Contoh Kamus Data

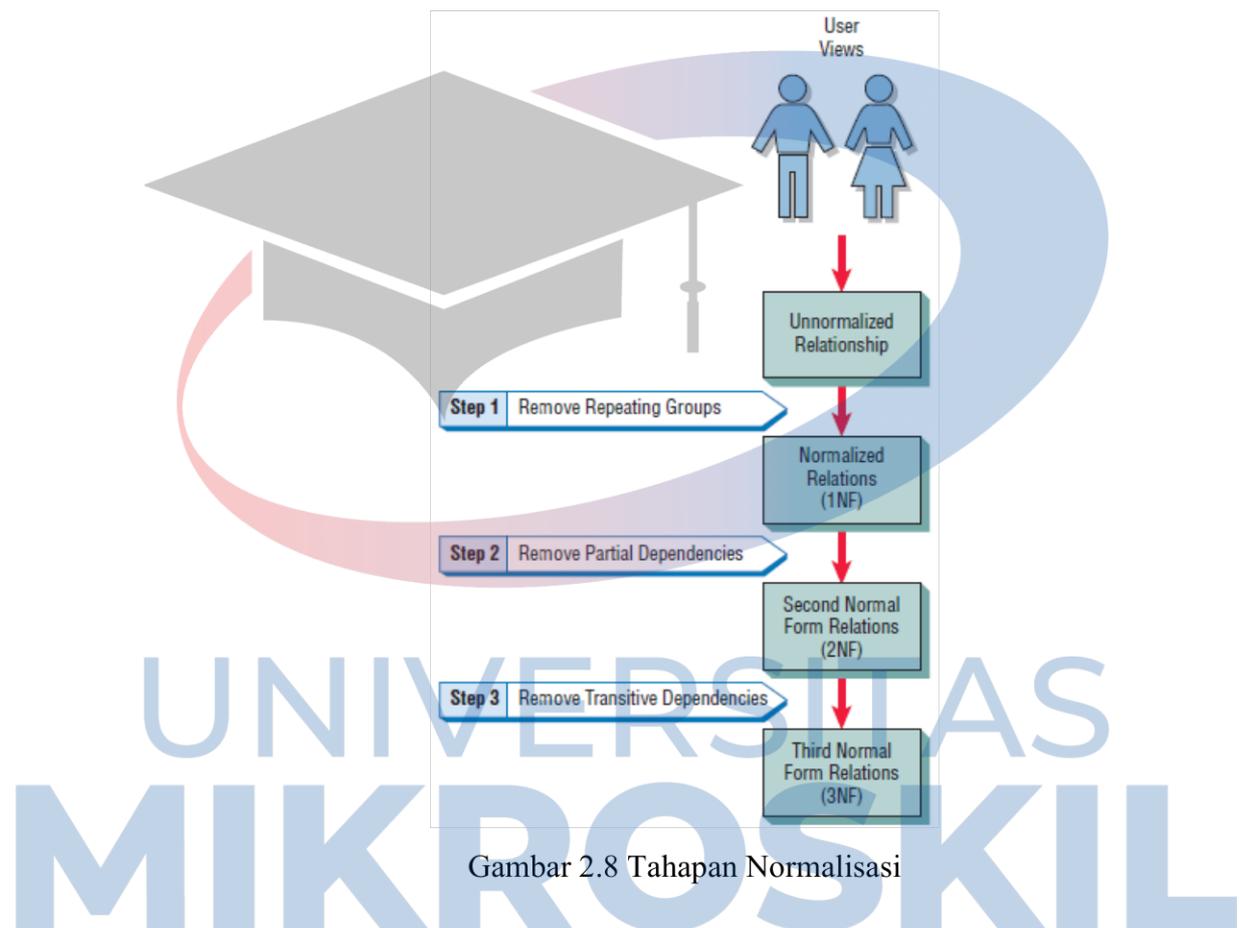
2.3.4 Normalisasi

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 [3].

Hubungan diperoleh dari tinjauan pemakai atau data tersimpan sebagian besar akan menjadi tidak normal. 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 mentransformasikan hubungan ke bentuk normalisasi ketiga [3].

Tahap kedua menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain [3].

Tahap ketiga mengubah ketergantungan transitif manapun. Suatu ketergantungan transitif adalah sesuatu dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya [3].



Gambar 2.8 Tahapan Normalisasi

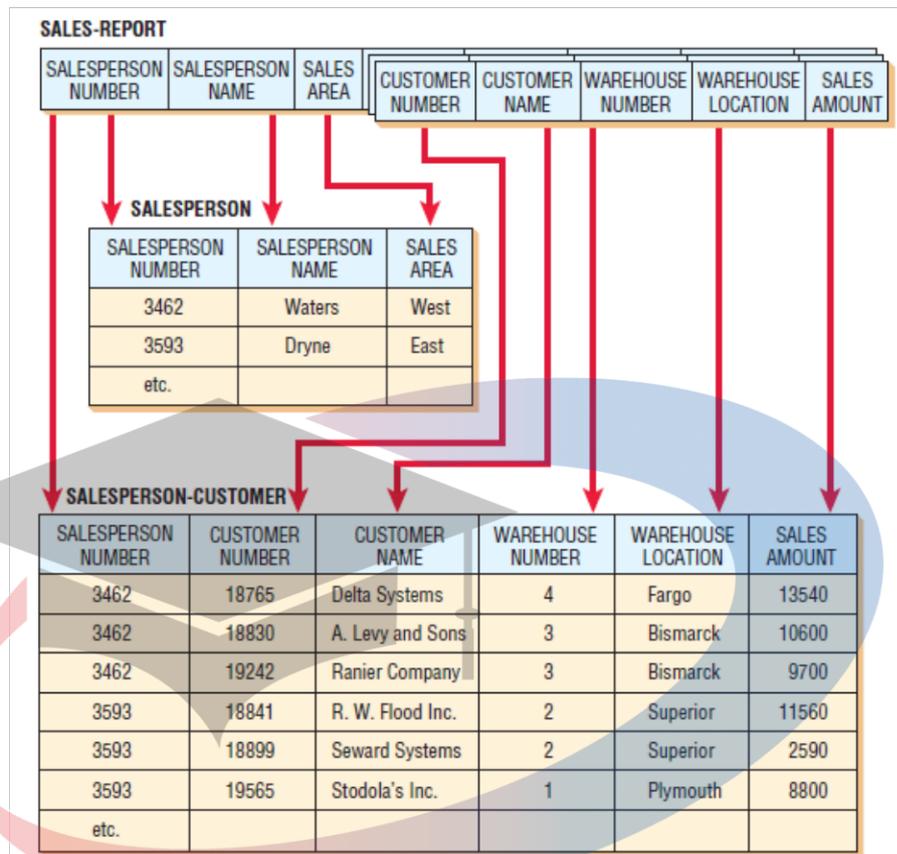
Berikut ini adalah contoh normalisasi dari suatu tabel [3].

| 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 2.9 Tabel yang Belum dinormalisasikan

Gambar di atas menunjukkan hubungan tidak normal LAPORAN-PENJUALAN yang akan dinormalisasikan dengan memisah hubungan ke dalam dua hubungan baru. Hubungan SALES mengandung kunci utama NOMOR-SALES dan semua atribut yang tidak terulang (NAMA-SALES dan DAERAH-PENJUALAN). Hubungan kedua, PELANGGAN-SALES, mengandung kunci utama dari hubungan SALES (kunci utama dari SALES adalah NOMOR-SALES) dan semua atribut yang berulang. Dengan mengetahui NOMOR-SALES secara otomatis, maka akan diketahui NAMA-PELANGGAN, JUMLAH-PENJUALAN, LOKASI GUDANG, dan lain sebagainya. Dalam hubungan ini, gunakan sebuah kunci gabungan untuk NOMOR-SALES dan NOMOR-PELANGGAN untuk mengakses informasi, dimana hubungan dapat digambarkan sebagai berikut [3].

UNIVERSITAS
MIKROSKIL



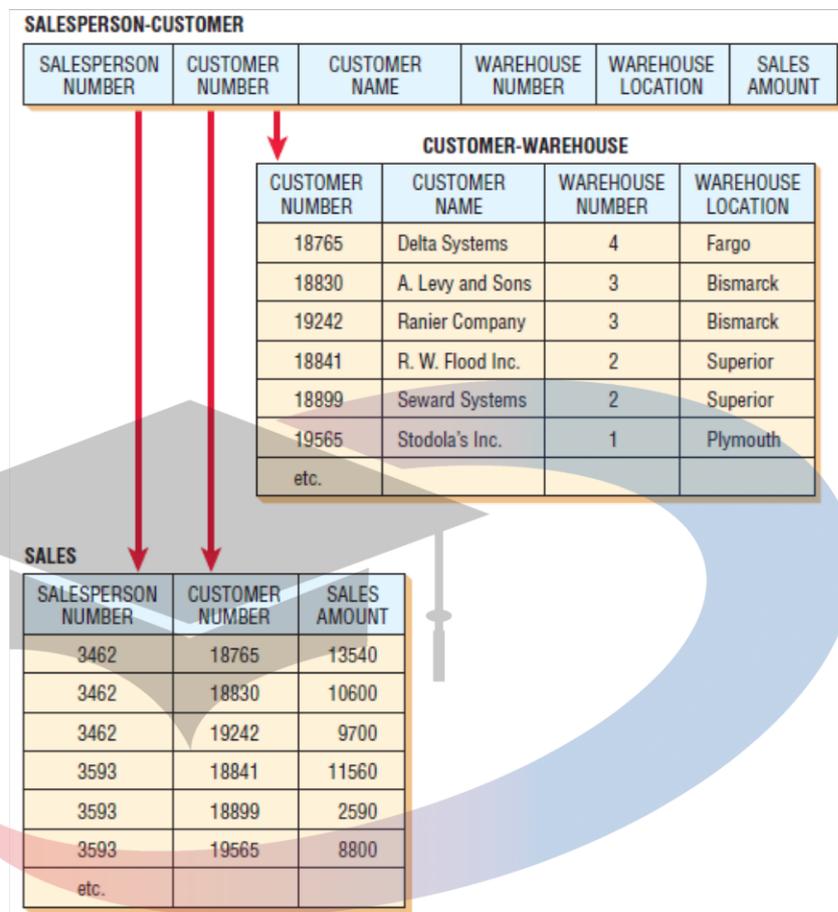
Gambar 2.10 Bentuk Normalisasi Pertama (1NF)

Dan hubungan dapat ditulis secara singkat sebagai berikut [3]:

SALES (NOMOR-SALES, NAMA-SALES, DAERAH-PENJUALAN) (3NF)
dan

PELANGGAN-SALES (NOMOR-SALES, NOMOR-PELANGGAN, NOMOR-GUDANG, LOKASI GUDANG, JUMLAH PENJUALAN) (1NF)

Dari hasil normalisasi di atas, ada beberapa atribut PELANGGAN-SALES yang tidak tergantung secara fungsional pada kunci utama. Atribut JUMLAH-PENJUALAN tergantung kepada dua kunci, sedangkan atribut NAMA-PELANGGAN, NOMOR-GUDANG, dan LOKASI-GUDANG hanya tergantung pada NOMOR-PELANGGAN, sehingga PELANGGAN-SALES dipisah menjadi dua hubungan baru yang digambarkan sebagai berikut [3].



Gambar 2.11 Bentuk Normalisasi Kedua (2NF)

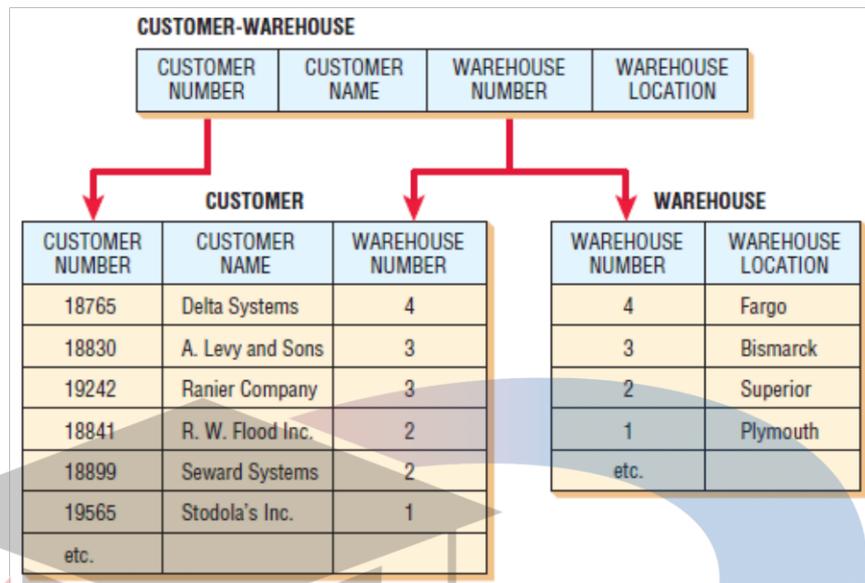
Hubungan dapat ditulis secara singkat sebagai berikut [3]:

PENJUALAN (NOMOR-SALES, NOMOR-PELANGGAN, JUMLAH-PENJUALAN) (1NF)

dan

GUDANG PELANGGAN (NOMOR-PELANGGAN, NAMA-PELANGGAN, NOMOR-GUDANG, LOKASI-GUDANG) (2NF)

Untuk hubungan normalisasi kedua, semua atribut harus bergantung pada kunci utama NOMOR-PELANGGAN, tetapi LOKASI-GUDANG juga masih tergantung pada NOMOR-GUDANG. Oleh karena itu, untuk menyederhanakan hubungan diperlukan langkah lain. Pada tahap ini juga dilakukan penguraian dari hubungan GUDANG-PELANGGAN [3].



Gambar 2.12 Bentuk Normalisasi Ketiga

Hubungan dapat ditulis secara singkat sebagai berikut [3]:

PELANGGAN (NOMOR-PELANGGAN, NAMA-PELANGGAN, NOMOR-GUDANG) (1NF)

dan

GUDANG (NOMOR-GUDANG, LOKASI GUDANG) (1NF)

Selain kunci utama, identifikasikan NOMOR-GUDANG menjadi kunci asing dalam hubungan PELANGGAN.

2.4 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. Orang yang memastikan bahwa basis data memenuhi tujuannya disebut administrator basis data [3].

Tujuan basis data yang efektif, yaitu [3]:

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.

Tujuan yang telah disebutkan di atas mengingatkan akan keuntungan dan kerugian pendekatan basis data. Pertama, pemakaian data berarti bahwa data hanya perlu disimpan sekali. Membantu mencapai integritas data, karena mengubah data yang diselesaikan lebih mudah dan dapat dipercaya jika data hanya muncul sekali dalam banyak *file* berbeda [3].

Ketika pemakai memerlukan data khusus, basis data yang dirancang dengan baik akan lebih memenuhi kebutuhan data yang dibutuhkan. Basis data yang dirancang dengan baik juga lebih fleksibel daripada *file* terpisah. Oleh karena itu, basis data dapat berkembang seiring dengan perubahan pada kebutuhan pemakai dan aplikasinya [3].

Sebuah bahasa basis data biasanya dapat dipisah ke dalam dua bentuk, yaitu [8]:

1. *Data Definition Language* (DDL)

Data Definition Language (DDL) adalah sebuah bahasa yang mengizinkan *Database Administrator* (DBA) atau pengguna untuk menjelaskan dan memberi nama pada entitas, atribut, dan hubungan yang diperlukan untuk aplikasi, secara bersamaan dengan *associated integrity* dan *security constraint*. DDL mengizinkan pengguna untuk menentukan tipe, struktur, serta kendala data yang nantinya akan disimpan ke dalam basis data.

2. *Data Manipulation Language* (DML)

Data Manipulation Language (DML) adalah sebuah bahasa yang menyediakan satu *set* operasi untuk mendukung dasar dari operasi manipulasi pada data yang disimpan pada basis data. DML mengizinkan pengguna untuk menambah, mengubah, menghapus, dan mengambil data dari basis data tersebut. Bahasa standar dari DBMS ialah *Structured Query Language* (SQL).

Komponen utama dalam DBMS adalah sebagai berikut [3]:

1. Data

Data di dalam sebuah basis data dapat disimpan secara terintegrasi dan data dapat dipakai secara bersama-sama.

2. Perangkat keras (*Hardware*)

Terdiri dari semua peralatan komputer yang digunakan untuk pengolahan system basis data, berupa peralatan untuk penyimpanan basis data (*secondary storage* seperti *disk*), peralatan *input* dan *output*, serta peralatan komunikasi data.

3. Perangkat lunak (*Software*)

Berfungsi sebagai perantara (*interface*) antara pemakai dengan data fisik pada basis data. *Software* pada basis data dapat berupa:

- a. *Database Management System* (DBMS) yang menangani akses terhadap basis data sehingga pemakai tidak perlu memikirkan proses penyimpanan dan pengolahan data secara detail.
- b. Program-program aplikasi dan prosedur-prosedur.

4. Pemakai (*User*)

Database Administrator (DBA) merupakan orang atau tim yang bertugas mengelola sistem basis data secara keseluruhan. DBA mempunyai tugas mengontrol DBMS dan *software-software*, memonitor siapa yang mengakses basis data, mengatur pemakaian basis data, memeriksa keamanan, integrasi, *recovery*, atau cadangan data, serta persetujuan.

- a. *Programmer*, merupakan orang atau tim yang bertugas membuat program aplikasi, misalnya untuk perbankan atau administrasi.
- b. *End-user*, merupakan orang yang mengakses basis data melalui terminal dengan menggunakan bahasa *query* atau program aplikasi yang dibuat oleh *programmer*.

2.5 Penjualan

Penjualan adalah kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli [8].

Secara umum penjualan terdiri dari dua jenis, yaitu penjualan tunai dan penjualan kredit. Penjualan tunai dilakukan oleh perusahaan dengan mewajibkan

pembeli melakukan pembayaran harga produk terlebih dahulu sebelum produk diserahkan oleh perusahaan kepada pembeli. Setelah uang diterima oleh perusahaan, maka produk kemudian diserahkan kepada pembeli dan transaksi penjualan tunai kemudian dicatat oleh perusahaan [8].

Penjualan tunai terjadi jika penyerahan barang atau jasa segera diikuti dengan pembayaran dari pembelian, sedangkan penjualan kredit terdapat tenggang waktu antara penyerahan barang atau jasa dalam penerimaan pembelian. Dalam penjualan kredit, pada saat penyerahan barang atau jasa, penjual menerima tanda bukti penerimaan barang. Keuntungan dari penjualan tunai adalah hasil dari penjualan tersebut langsung terealisasi dalam bentuk kas yang dibutuhkan perusahaan [8].

2.6 Pembelian

Pembelian merupakan kegiatan menukarkan uang sebagai alat transaksi yang sah dengan barang yang dilakukan oleh dua orang atau lebih. Pada perusahaan dagang, pembelian ini dilakukan untuk mendapatkan barang dagangan atau persediaan barang dagangan, yang nantinya akan dijual kembali kepada konsumen [9].

Fungsi pembelian sebagai berikut [9]:

1. Pembelian atas barang dagangan (untuk perusahaan dagang dan bahan baku, bahan setengah jadi, suku cadang untuk perusahaan manufaktur).
2. Pembelian aktiva tetap, seperti mesin-mesin, peralatan kantor
3. Pembelian barang pendukung produksi
4. Pembelian barang - barang keperluan perusahaan
5. Menjalinkan hubungan dengan *supplier* barang
6. Menerbitkan *order* barang dagangan
7. Memberikan verifikasi atau pembuktian pada *supplier* tentang kebenaran pembelian yang telah dilakukan.

2.7 Persediaan

Persediaan adalah aset lancar berupa barang jadi yang disimpan untuk dijual dalam kegiatan normal perusahaan dan bahan yang diproses dalam proses produksi atau bahan yang disimpan untuk produksi. Persediaan di perusahaan dagang hanya diklasifikasikan sebagai Persediaan Barang Dagang (*Merchandise Inventory*),

sedangkan di perusahaan manufaktur, persediaan diklasifikasikan menjadi tiga, yaitu Persediaan Barang Jadi (*Finished Goods*), Persediaan Barang Dalam Proses (*Work In Process*), dan Persediaan Bahan Baku (*Materials*) [10].

Berdasarkan fungsinya, persediaan dikelompokkan menjadi [11]:

1. *Lot-Size-Inventory*, yaitu persediaan yang diadakan dalam jumlah lebih besar dari jumlah yang dibutuhkan saat itu. Cara ini dilakukan dengan tujuan memperoleh potongan harga karena pembelian dalam jumlah yang besar dan memperoleh biaya pengangkutan per unit yang rendah.
2. *Fluctuation stock*, yaitu persediaan yang diadakan untuk menghadapi permintaan yang tidak bisa diramalkan sebelumnya, serta untuk mengatas berbagai kondisi yang tidak terduga, seperti terjadi kesalahan dalam peramalan penjualan, kesalahan waktu produksi maupun kesalahan dalam pengiriman.
3. *Anticipation stock*, yaitu persediaan yang diadakan untuk menghadapi fluktuasi permintaan yang dapat diramalkan seperti mengantisipasi pengatur musim, yaitu ketika permintaan tinggi namun perusahaan tidak mampu memenuhinya. Di samping itu juga persediaan ini ditujukan untuk mengantisipasi kemungkinan sulitnya memperoleh bahan sehingga tidak mengganggu operasi perusahaan.

Perusahaan menggunakan salah satu dari dua jenis sistem agar pencatatan persediaan tetap akurat, yaitu sistem perpetual atau sistem periodik [10]:

1. Sistem perpetual, secara terus menerus melacak perubahan akun persediaan. Semua pembelian dan penjualan barang langsung dicatat ke akun persediaan pada saat transaksi terjadi.
2. Sistem periodik, kuantitas persediaan yang masih tersedia ditentukan secara periodik. Dalam sistem periodik, perhitungan fisik persediaan perlu dilakukan setidaknya sekali setahun pada setiap akhir tahun.