

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah suatu kumpulan objek atau unsur-unsur atau bagian-bagian yang memiliki arti berbeda-beda yang saling memiliki hubungan, saling berkerjasama dan saling memengaruhi satu sama lain serta memiliki keterikatan pada rencana atau plane yang sama dalam mencapai suatu tujuan tertentu pada lingkungan yang kompleks [3].

Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut [4]:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang berkerjasama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara.

Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem tersebut yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolah Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.2 Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah kejadian yang terjadi pada saat tertentu. Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi, karena informasi berguna memberikan gambaran tentang suatu permasalahan sehingga pengambil keputusan dapat menentukan keputusan lebih cepat, informasi juga memberikan standar, aturan maupun indicator bagi pengambil keputusan. [5]

Kualitas suatu informasi tergantung dari 3 (tiga) hal, yaitu: informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*). Penjelasan tentang kualitas informasi tersebut akan dipaparkan di bawah ini [4].

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena biasanya dari sumber informasi sampai penerima informasi ada kemungkinan terjadi gangguan (*noise*) yang dapat mengubah atau merusak informasi tersebut.

2. Tepat waktu (*timelines*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal bagi organisasi. Dewasa ini, mahalnya informasi disebabkan karena harus cepatnya informasi tersebut dikirim atau didapat sehingga diperlukan teknologi mutakhir untuk mendapat, mengolah, dan mengirimkannya.

3. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk orang satu dengan yang lain berbeda, misalnya informasi sebab musibah kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan apabila ditujukan kepada ahli teknik perusahaan. Sebaliknya, informasi mengenai harga pokok produksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi akan sangat relevan untuk seorang akuntan perusahaan.

Nilai informasi ini didasarkan atas 10 sifat yaitu:

a. Mudah diperoleh

Sifat ini menunjukkan kemudahan dan kecepatan untuk memperoleh Informasi.

b. Luas dan lengkap

Sifat ini menunjukkan kelengkapan isi Informasi. Hal ini tidak hanya mengenai volumenya, akan tetapi juga mengenai keluaran Informasi.

c. Ketelitian

Sifat ini berhubungan dengan tingkat kebebasan dari kesalahan keluaran Informasi.

d. Kecocokan

Sifat ini menunjukkan seberapa baik keluaran Informasi dalam hubungannya dengan permintaan para pemakai.

e. Ketepatan waktu

Sifat ini berhubungan dengan waktu yang dilalui, lebih pendek dari siklus untuk mendapatkan Informasi.

f. Kejelasan

Sifat ini menunjukkan tingkat kejelasan Informasi, Informasi hendaknya terbebas dari istilah-istilah yang tidak jelas.

g. Keluwesan

Sifat ini berhubungan dengan apakah Informasi tersebut dapat digunakan untuk membuat lebih dari suatu keputusan, tetapi juga apakah dapat digunakan untuk lebih dari seseorang pengambil keputusan.

h. Dapat dibuktikan

Sifat ini menunjukkan sejauh mana Informasi itu dapat diuji oleh beberapa pemakai hingga sampai didapatkan kesimpulan yang sama.

i. Tidak ada prasangka

Sifat ini berhubungan dengan ada tidaknya keinginan untuk mengubah Informasi tersebut guna mendapatkan kesimpulan yang telah diarahkan sebelumnya.

j. Dapat diukur

Sifat ini menunjukkan hakikat Informasi yang dihasilkan oleh sistem Informasi formal.

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

2.1.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi

operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [4].

Sistem Informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yaitu [4] :

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem Informasi. Yang dimaksud dengan input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem adalah keluaran yang merupakan Informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi merupakan *tool box* dalam sistem Informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan.

5. Blok basis data (*database block*)

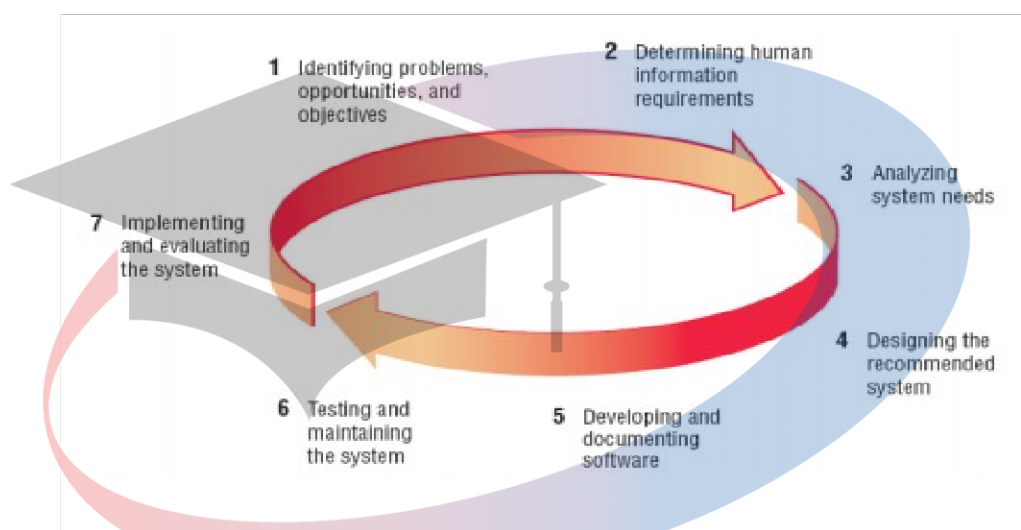
Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya.

6. Blok kendali (*control block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangankecurangan, kegagalan pada sistem itu sendiri dan lain sebagainya. Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi.

2.2 System Development Life Cycle (SDLC)

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 [6].



Gambar 2. 1 Siklus Hidup Pengembangan Sistem

Metodologi ini terdiri dari 7 (tujuh) tahapan yang dijabarkan sebagai berikut [6]:

1. Mengidentifikasi masalah, peluang, dan tujuan

Di tahap pertama dari siklus hidup pengembangan sistem ini, penganalisis mengidentifikasi masalah, peluang, dan tujuan-tujuan yang hendak dicapai. Penganalisis melihat dengan jujur pada apa yang terjadi di dalam bisnis. Kemudian, bersama-sama dengan anggota organisasional lain, penganalisis menentukan dengan tepat masalah-masalah tersebut. Peluang adalah situasi di mana penganalisis yakin bahwa peningkatan bisa dilakukan melalui penggunaan sistem informasi terkomputerisasi. Mengukur peluang memungkinkan bisnis untuk mencapai sisi kompetitif atau menyusun standar-standar industri.

Penganalisis harus menemukan apa yang sedang dilakukan dalam bisnis. Barulah kemudian penganalisis akan bisa melihat beberapa aspek dalam aplikasi-aplikasi

sistem informasi untuk membantu bisnis supaya mencapai tujuan-tujuannya dengan menyebut problem atau peluang-peluang tertentu.

2. Menentukan syarat-syarat informasi

Dalam tahap berikutnya, penganalisis memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Dalam tahap syarat-syarat informasi siklus hidup pengembangan sistem, penganalisis berusaha keras untuk memahami informasi apa yang dibutuhkan pemakai agar bisa ditampilkan dalam pekerjaan mereka. Penganalisis sistem perlu tahu detail-detail fungsi-fungsi sistem yang ada: siapa (orang-orang yang terlibat), apa (kegiatan bisnis), dimana (lingkungan dimana pekerjaan itu dilakukan), kapan (waktu yang tepat), dan bagaimana (bagaimana prosedur yang harus dijalankan) dari bisnis yang sedang dipelajari.

3. Menganalisis kebutuhan sistem

Perangkat dan teknik-teknik tertentu akan membantu penganalisis menentukan kebutuhan yaitu 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 berisikan daftar seluruh item data yang digunakan dalam sistem, berikut spesifiknya apakah berupa *alphanumeric* atau teks, serta berapa banyak spasi yang dibutuhkan saat dicetak.

4. Merancang sistem yang direkomendasikan

Penganalisis merancang prosedur *data-entry* sedemikian rupa sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Selain itu, penganalisis menggunakan teknik-teknik bentuk dan perancangan layar tertentu untuk menjamin keefektifan input sistem informasi. Tahap perancangan juga mencakup perancangan *file-file* atau basis data yang bisa menyimpan data-data yang diperlukan oleh pembuat keputusan. Basis data yang tersusun dengan baik adalah dasar bagi seluruh sistem informasi. Dalam tahap ini, penganalisis juga bekerja sama dengan pemakai untuk merancang output (baik pada layar maupun hasil cetakan). Terakhir, penganalisis harus merancang prosedur-prosedur *back up* dan *control* untuk melindungi sistem dan data serta untuk membuat paket-paket spesifikasi program bagi pemrogram.

5. Mengembangkan dan mendokumentasikan perangkat lunak

Penganalisis bekerja bersama-sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang diperlukan. Beberapa teknik terstruktur untuk merancang dan mendokumentasikan perangkat lunak meliputi rencana terstruktur, *Nassi-Shneiderman charts* dan *pseudocode*. Penganalisis sistem menggunakan salah satu semua perangkat ini untuk memprogram apa yang perlu diprogram.

6. Menguji dan mempertahankan sistem

Sebelum sistem informasi dapat digunakan, maka harus dilakukan pengujian terlebih dulu. 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.

7. Mengimplementasikan dan mengevaluasi sistem

Di tahap terakhir dari pengembangan sistem, penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan 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 pengubahan file-file dari format lama ke format baru atau membangun suatu basis data, meng-install peralatan dan membawa sistem baru untuk diproduksi. Evaluasi yang ditunjukkan sebagai bagian dari tahap terakhir dari siklus hidup pengembangan sistem biasanya dimaksudkan untuk pembahasan. Kriteria utama yang harus dipenuhi ialah apakah pemakai yang dituju benar-benar menggunakan sistem.

2.3 PIECES

PIECES adalah kerangka yang dipakai untuk mengklasifikasikan suatu *problem*, *opportunities* dan *directives* yang terdapat pada bagian *scope definition* analisis dan perancangan sistem [7].

Dalam *PIECES* terdapat enam buah variabel yang digunakan untuk menganalisis sistem informasi, yaitu [7]:

1. *Performance* (Keandalan)

Analisis ini dilakukan untuk mengetahui kinerja sebuah sistem, apakah berjalan dengan baik atau tidak. Kinerja ini dapat diukur dari jumlah temuan data yang dihasilkan dan seberapa cepat suatu data dapat ditemukan.

2. *Information and Data* (Data dan Informasi)

Dalam sebuah temuan data pasti akan dihasilkan sebuah informasi yang akan ditampilkan, analisis ini digunakan untuk mengetahui seberapa banyak dan seberapa jelas informasi yang akan dihasilkan untuk satu pencarian.

3. *Economics* (Nilai Ekonomis)

Analisis ini dilakukan untuk mengetahui apakah suatu sistem itu tepat diterapkan pada suatu lembaga informasi dilihat dari segi finansial dan biaya yang dikeluarkan. Hal ini sangat penting karena suatu sistem juga dipengaruhi oleh besarnya biaya yang dikeluarkan.

4. *Control and Security* (Pengendalian dan Pengamanan)

Dalam suatu sistem perlu diadakan sebuah kontrol atau pengawasan agar sistem itu berjalan dengan baik. Analisis ini digunakan untuk mengetahui sejauh mana pengawasan dan kontrol yang dilakukan agar sistem tersebut berjalan dengan baik.

5. *Efficiency* (Efisiensi)

Efisiensi dan efektivitas sebuah sistem perlu dipertanyakan dalam kinerja dan alasan mengapa sistem itu dibuat. Sebuah sistem harus bisa secara efisien menjawab dan membantu suatu permasalahan khususnya dalam hal otomasi. Analisis ini dilakukan untuk mengetahui apakah suatu sistem itu efisien atau tidak, dengan input yang sedikit bisa menghasilkan sebuah *output* yang memuaskan.

6. *Service* (Pelayanan)

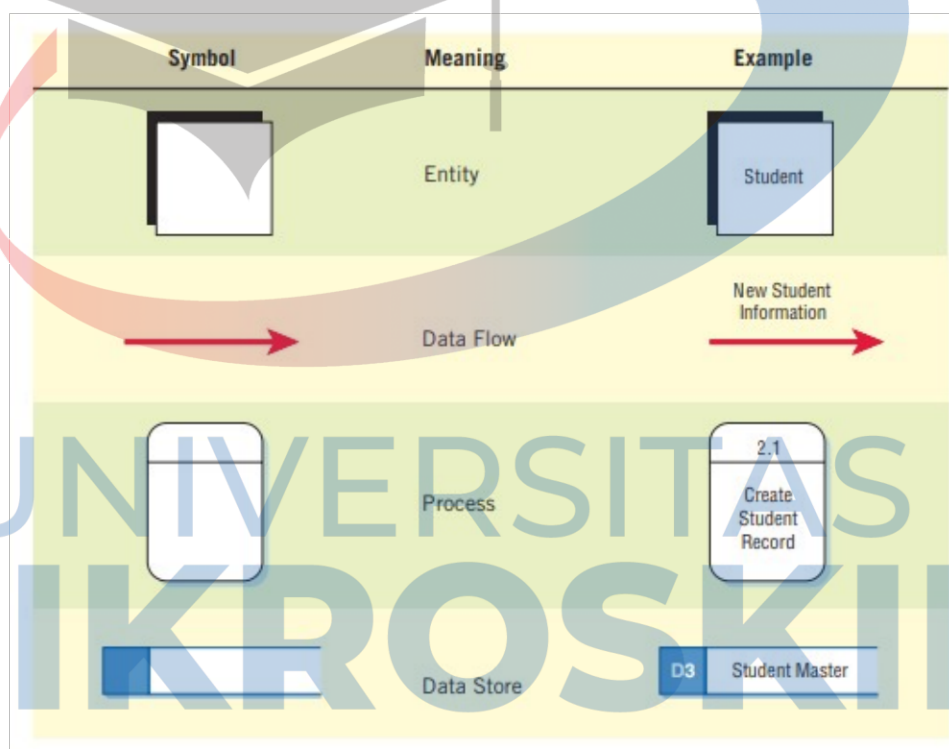
Dalam hal pemanfaat suatu sistem, sebuah pelayanan masih menjadi suatu hal yang penting dan perlu diperhatikan. Suatu sistem yang diterapkan akan berjalan dengan baik dan seimbang bila diimbangi dengan pelayanan yang baik juga. Analisis ini digunakan untuk mengetahui bagaimana pelayanan yang dilakukan dan mengetahui permasalahan-permasalahan yang ada terkait tentang pelayanan.

2.4 Teknik Pengembangan Sistem

2.4.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafis dari aliran data melalui sistem informasi. Hal ini memungkinkan untuk mewakili proses dalam sistem informasi dari sudut pandang data. *DFD* memungkinkan untuk memvisualisasikan bagaimana sistem beroperasi, apa sistem menyelesaikan dan bagaimana itu akan dilaksanakan, bila disempurnakan dengan spesifikasi lebih lanjut. *Data flow diagram* digunakan oleh analis sistem untuk merancang sistem pemrosesan informasi tetapi juga sebagai cara untuk model seluruh organisasi [8].

Empat simbol dasar yang digunakan dalam diagram aliran data, yaitu [6] :



Gambar 2. 2 Contoh Simbol Dasar dalam DFD

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 diagram aliran data adalah [6]:

1. Kotak Rangkap Dua

Untuk menggambarkan suatu entitas eksternal (bagian lain, sebuah perusahaan, seseorang, atau sebuah mesin) yang dapat mengirim data atau menerima data dari sistem.

2. Tanda Panah

Untuk menunjukkan perpindahan data dari satu titik ke titik yang lain, dengan kepala tanda panah mengarah ke tujuan data.

3. Bujur Sangkar Dengan Sudut Membulat

Untuk menunjukkan adanya proses transformasi.

4. Bujur Sangkar

Dengan ujung terbuka (bujur sangkar yang digambarkan dengan dua garis paralel yang tertutup oleh sebuah garis pendek di sisi kiri dan ujungnya terbuka di sisi sebelah kanan).

Beberapa kesalahan umum yang dibuat saat menggambar diagram aliran data adalah sebagai berikut [6] :

1. Lupa memasukkan aliran data atau mengarahkan panah ke arah yang salah. Contohnya adalah proses menggambar yang menunjukkan semua aliran datanya sebagai input atau output. Setiap proses mengubah data dan harus menerima masukan dan menghasilkan keluaran saja.
2. Menghubungkan penyimpanan data dan entitas eksternal secara langsung satu sama lain. Penyimpanan data dan entitas mungkin tidak terhubung satu sama lain; penyimpanan data dan entitas eksternal harus terhubung hanya dengan suatu proses.
3. Proses pelabelan atau aliran data yang salah. Periksa diagram aliran data untuk memastikan bahwa setiap objek atau aliran data diberi label dengan benar. Suatu proses harus menunjukkan nama sistem atau menggunakan format kata kerja-kata sifat-kata benda. Setiap aliran data harus dijelaskan dengan kata benda.
4. Memasukkan lebih dari sembilan proses pada diagram aliran data. Memiliki terlalu banyak proses membuat diagram berantakan yang membingungkan untuk dibaca dan menghalangi daripada meningkatkan komunikasi.
5. Mengabaikan aliran data. Periksa diagram untuk aliran linier, yaitu aliran data di mana setiap proses hanya memiliki satu masukan dan satu keluaran. Kecuali dalam kasus diagram aliran data anak yang sangat rinci, aliran data linier agak

jarang. Kehadirannya biasanya menunjukkan bahwa diagram memiliki aliran data yang hilang.

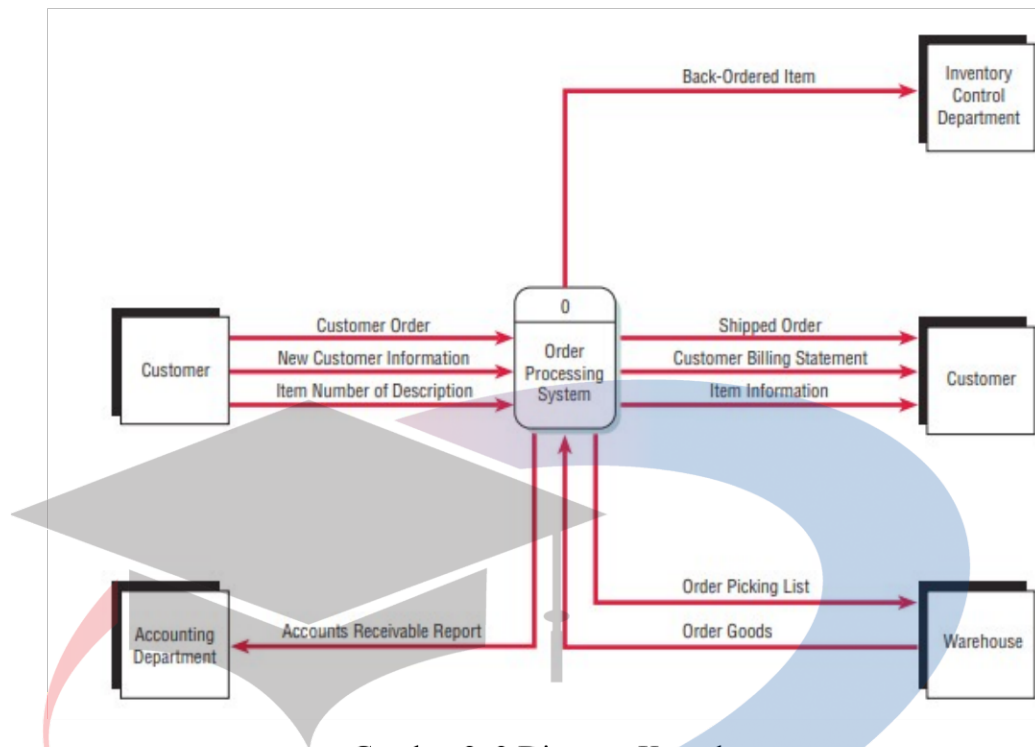
6. Membuat dekomposisi (atau ledakan) yang tidak seimbang dalam diagram anak. Setiap diagram anak harus memiliki aliran data masukan dan keluaran yang sama seperti proses induk. Pengecualian untuk aturan ini adalah keluaran minor, seperti garis kesalahan, yang hanya disertakan pada diagram anak.

Langkah-langkah menggambarkan diagram aliran data adalah sebagai berikut:

1. Merancang Diagram Konteks (*Context Diagram*)

Dengan pendekatan atas-bawah untuk membuat diagram pergerakan data, diagram berpindah dari umum ke spesifik. Meskipun diagram pertama membantu analis sistem memahami pergerakan data dasar, sifat umumnya membatasi kegunaannya. Diagram konteks awal harus berupa gambaran umum, yang mencakup masukan dasar, sistem umum, dan keluaran. Diagram ini akan menjadi diagram yang paling umum, benar-benar merupakan gambaran menyeluruh dari 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. Prosesnya diberi angka nol. Semua entitas eksternal ditampilkan pada diagram konteks, serta aliran data utama ke dan dari mereka. Diagram tidak berisi penyimpanan data apa pun dan cukup sederhana untuk dibuat, setelah entitas eksternal dan aliran data ke dan dari mereka diketahui analis.

Berikut ini adalah contoh penggambaran diagram konteks [6] :

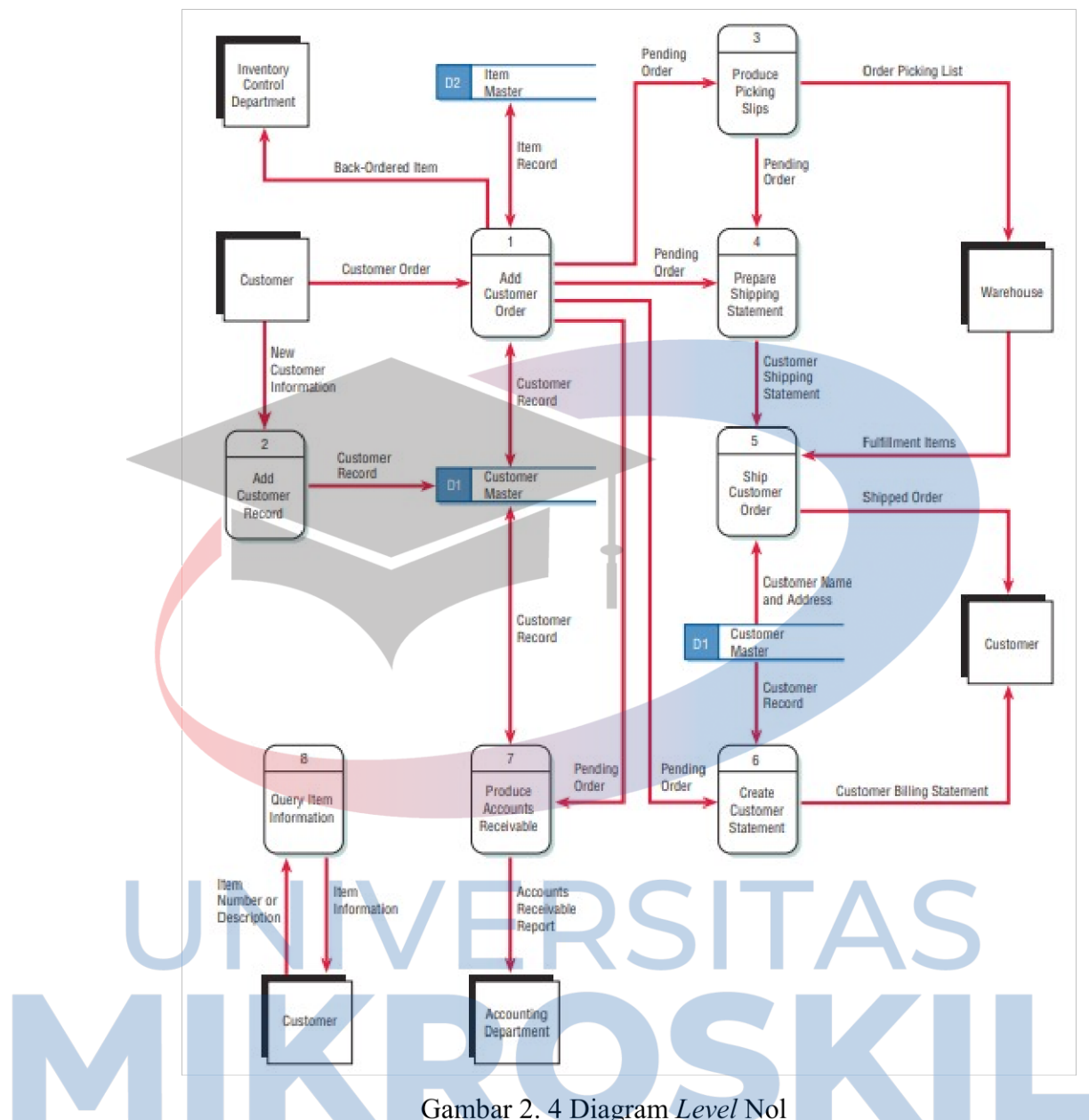


Gambar 2. 3 Diagram Konteks

2. Merancang Diagram Nol

Masukan dan keluaran yang ditentukan pada diagram pertama tetap konstan di semua diagram berikutnya. 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 menghasilkan diagram yang berantakan yang sulit dipahami. Setiap proses diberi nomor dengan bilangan bulat, umumnya dimulai dari sudut kiri atas diagram dan bekerja ke sudut kanan bawah. Penyimpanan data utama dari sistem (mewakili *file master*) dan semua entitas eksternal disertakan pada Diagram nol.

Berikut ini adalah contoh penggambaran diagram *level* nol [6] :

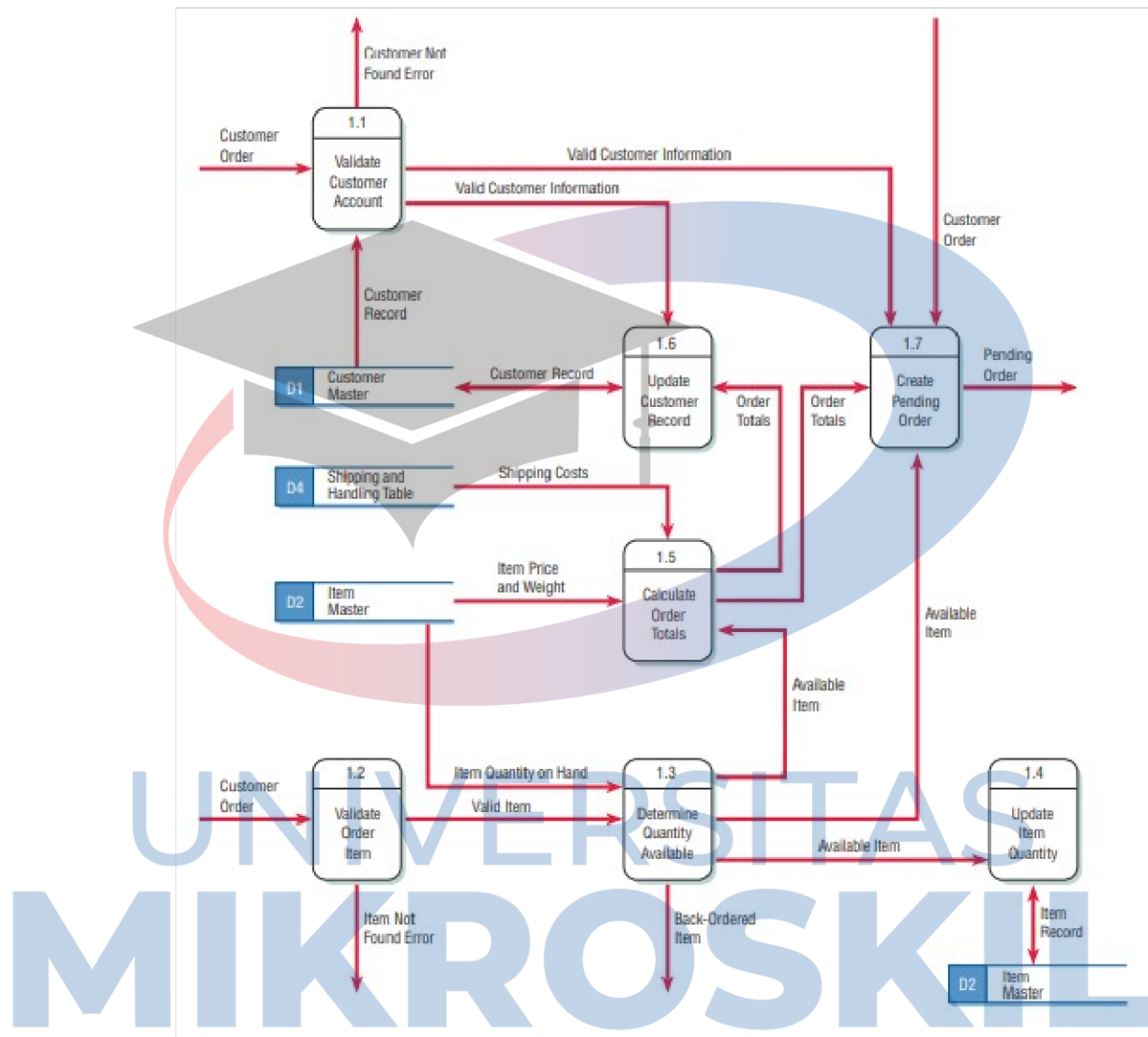
Gambar 2. 4 Diagram *Level Nol*

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

Setiap proses pada Diagram 0 bisa dikembangkan untuk membuat diagram anak yang lebih detail. Proses pada Diagram 0 yang meledak disebut proses induk, dan diagram yang dihasilkan disebut diagram anak. Aturan utama untuk membuat diagram anak, keseimbangan vertikal, menyatakan bahwa diagram anak tidak dapat menghasilkan keluaran atau menerima masukan yang tidak juga diproduksi atau diterima oleh proses induk. Semua aliran data masuk atau keluar

dari proses induk harus ditampilkan mengalir masuk atau keluar dari diagram anak.

Berikut ini adalah contoh penggambaran diagram *level nol* [6]:

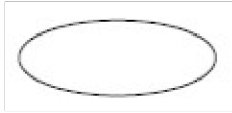
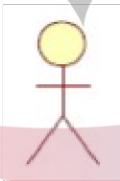






Gambar 2. 5 Diagram Anak

2.4.2 Use Case

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu [9]:

Tabel 2. 1 Simbol Use Case Diagram

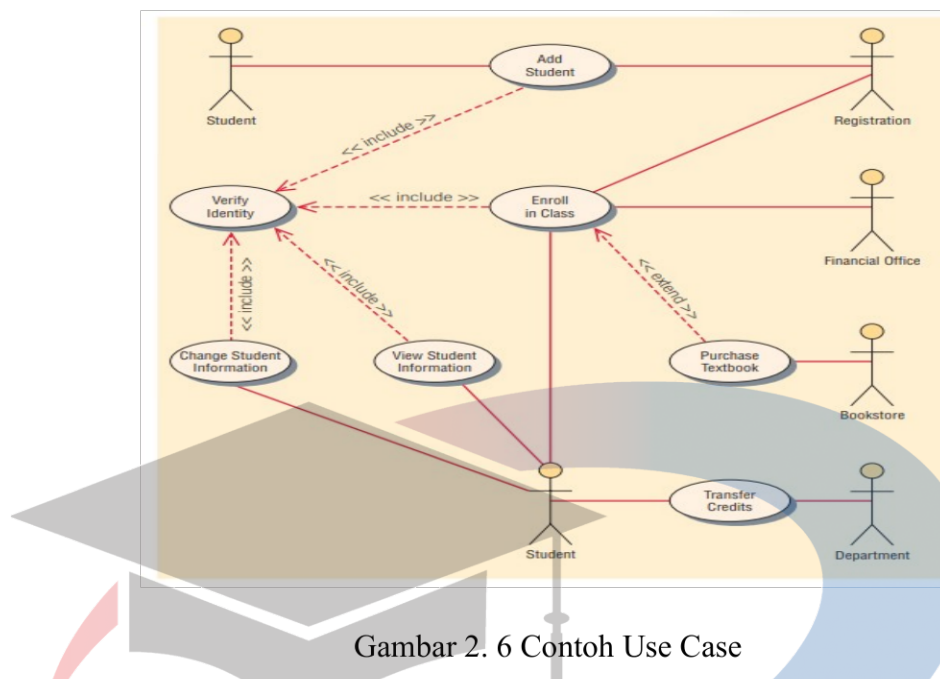
Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar <i>unit</i> dengan aktor, yang dinyatakan dengan menggunakan kata kerja</p>
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>

	<p><i>Include</i> merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program</p>
	<p><i>Extend</i> merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

Berikut ini cara menggambar use case diagram yaitu [6] :

1. Meninjau spesifikasi bisnis dan identifikasi pelaku yang terlibat.
2. Mengidentifikasi peristiwa tingkat tinggi dan mengembangkann kasus penggunaan utama yang menggambarkan peristiwa tersebut dan bagaimana aktor memulainya. Memeriksa dengan cermat peran yang dimainkan oleh para aktor untuk mengidentifikasi semua kemungkinan kasus penggunaan utama yang diprakarsai oleh masing-masing aktor. Kasus penggunaan dengan sedikit atau tanpa interaksi pengguna tidak harus ditampilkan.
3. Meninjau setiap kasus penggunaan utama untuk menentukan kemungkinan variasi aliran melalui kasus penggunaan. Dari analisis ini, ditentukan jalur alternatif. Karena aliran peristiwa biasanya berbeda di setiap kasus, mencari kegiatan yang bisa berhasil atau gagal. Serta mencari setiap cabang dalam logika use case yang memungkinkan hasil yang berbeda.

Berikut adalah salah satu contoh *Use Case* [6] :



Gambar 2. 6 Contoh Use Case

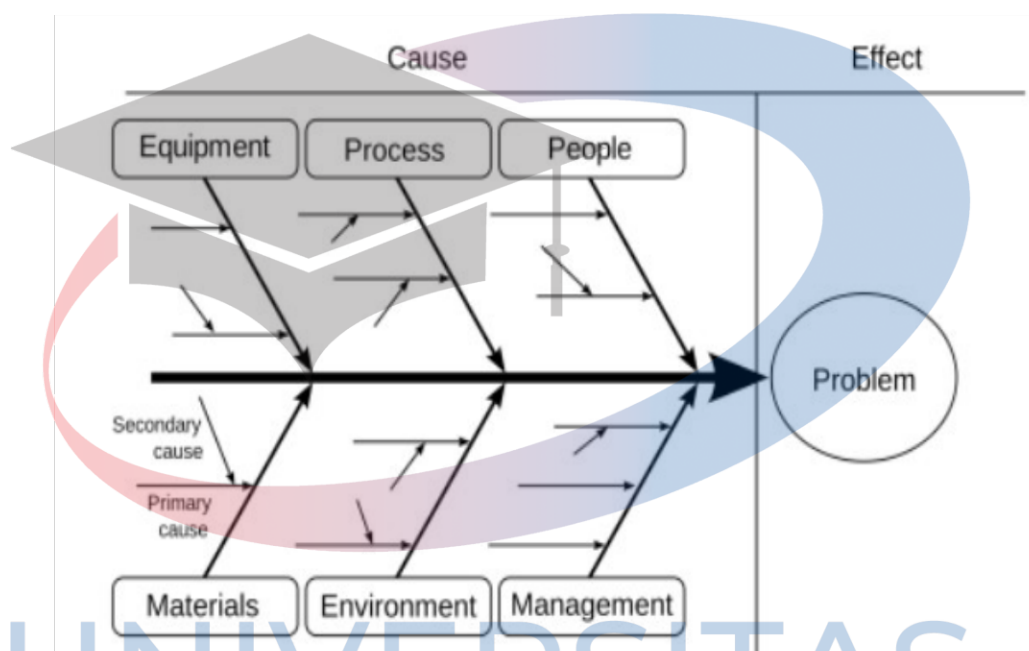
2.4.3 Diagram Ishikawa

Teknik analisa untuk mengidentifikasi sebab akibat dari permasalahan melalui diagram Ishikawa. Diagram Ishikawa atau *Fishbone* diagram (diagram tulang ikan) sering juga disebut *Cause-And-Effect* Diagram merupakan teknik untuk memetakan seluruh faktor yang menyebabkan terjadinya masalah pada hasil yang diinginkan. Adapun tujuan dari diagram Ishikawa adalah untuk mendata seluruh faktor yang mempengaruhi mutu dari sebuah proses dan untuk memetakan inter-relasi antar faktor-faktor. Semua kategori penyebab dimulai dengan 5M yaitu: manusia (*man*), mesin (*machine*), bahan (*material*), uang (*money*), dan metode [10].

Ishikawa menuturkan bahwa *Cause and Effect* Diagram digunakan tidak hanya untuk masalah *QC* akan tetapi dapat diterapkan untuk solusi dari masalah apa pun. Tujuannya adalah untuk mendapatkan hasil; karena tindakan harus diambil terhadap penyebabnya, jika kita tidak tahu hubungan antara sebab dan akibat dari suatu masalah, maka tidak akan dapat mengambil tindakan apapun untuk menyelesaikannya. *Cause and Effect* Diagram menunjukkan penyebab yang paling jelas sehingga dapat diambil tindakan dengan cepat. Diagram Ishikawa dirancang dalam konteks praktis dengan fokus pada pemahaman yang lebih dalam tentang “mengapa-nya” dalam proses [10].

Diagram Ishikawa dapat digunakan untuk memecah secara efektif masalah ke dalam matriks hubungan sebab akibat. Diagram tersebut digunakan untuk mengidentifikasi penyebab setelah difokuskan definisi masalah dalam tim atau *QC Circle*, beberapa individu dapat memberikan saran dan pendapat tentang faktor kausal apa saja yang paling berpengaruh yang menyebabkan efek atau masalah [10].

Berikut ini contoh Diagram Ishikawa [10]:



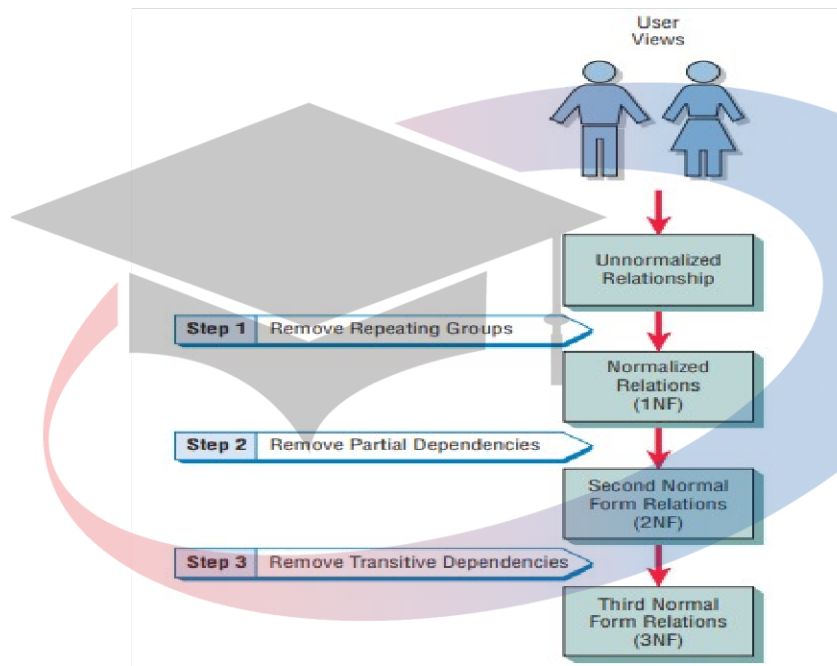
Gambar 2. 7 Contoh Diagram Ishikawa

2.4.4 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 [6].

Relasi yang diturunkan 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 utama. Untuk melakukannya, relasi perlu dipecah menjadi dua atau lebih relasi. Pada titik ini, relasi mungkin sudah dalam bentuk normal ketiga, tetapi kemungkinan lebih banyak langkah yang diperlukan untuk mengubah relasi ke bentuk normal ketiga. Langkah

kedua memastikan bahwa semua atribut *nonkey* sepenuhnya bergantung pada kunci utama. Semua dependensi parsial dihapus dan ditempatkan di relasi lain. Langkah ketiga menghilangkan ketergantungan transitif. Ketergantungan transitif adalah ketergantungan di mana atribut nonkunci bergantung pada atribut nonkunci lainnya [6].



Gambar 2. 8 Contoh Tahapan Normalisasi

UNIVERSITAS
MIKROSKIL

Berikut ini contoh dari normalisasi sebagai berikut [6]:

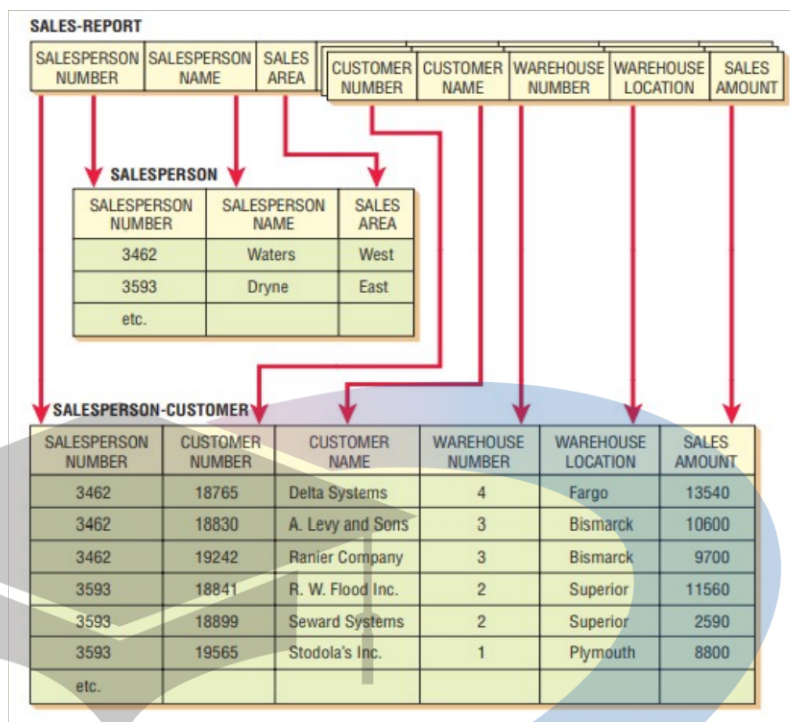
SALESPERSON NUMBER	SALESPERSON NAME	SALES AREA	CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES AMOUNT
3462	Waters	West	18765	Delta Systems	4	Fargo	13540
			18830	A. Levy and Sons	3	Bismarck	10600
			19242	Ranier Company	3	Bismarck	9700
3593	Dryne	East	18841	R. W. Flood Inc.	2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

Gambar 2. 9 Tabel Data Tidak Normal

Berdasarkan gambar diatas berikut ini tahapan normalisasi yang dilakukan [6]:

1. Bentuk Normalisasi Pertama (1NF)

UNIVERSITAS
MIKROSKIL



Gambar 2. 10 Contoh Bentuk Normalisasi Pertama (1NF)

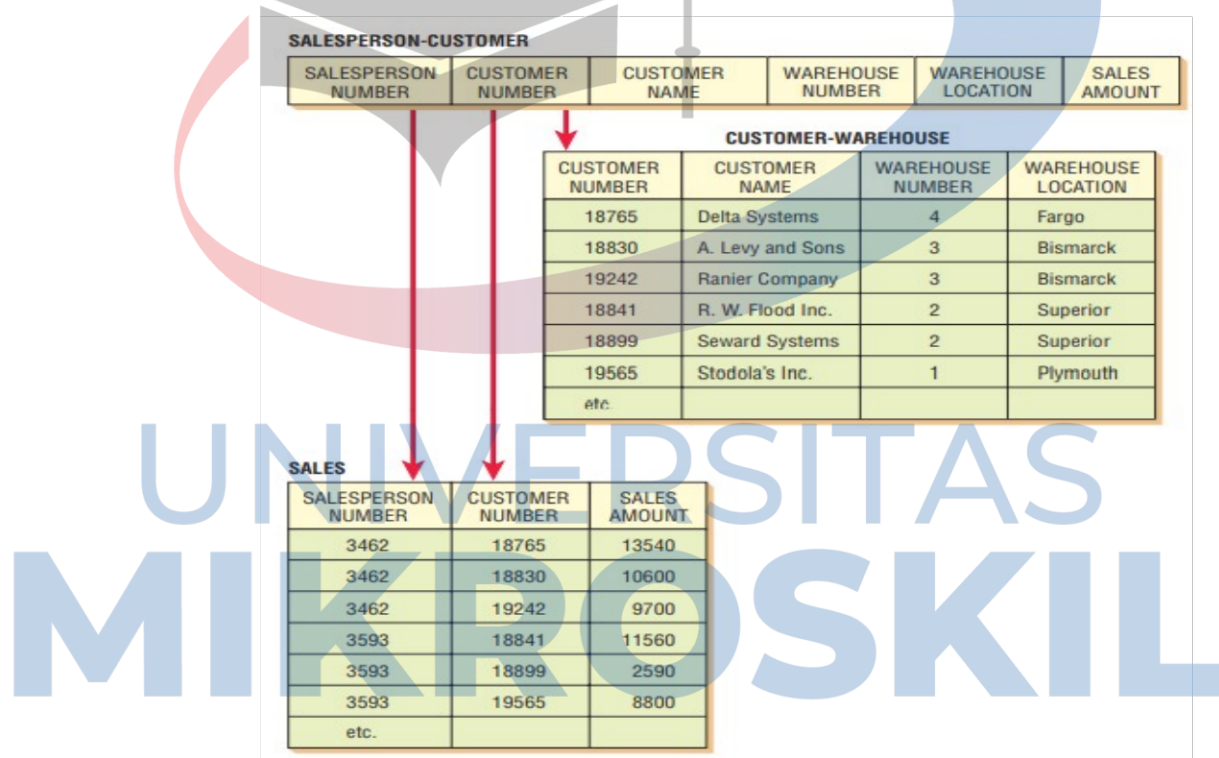
Langkah pertama dalam menormalkan relasi adalah menghapus grup yang berulang. Dalam contoh ini, relasi *SALES-REPORT* yang tidak dinormalisasi akan dipecah menjadi dua relasi terpisah. Relasi baru ini akan diberi nama *SALES PERSON* dan *SALES PERSON CUSTOMER*. Gambar diatas menunjukkan bagaimana relasi asli yang tidak dinormalisasi *SALES-REPORT* dinormalisasi dengan memisahkan relasi menjadi dua relasi baru. Perhatikan bahwa hubungan *SALES PERSON* berisi kunci utama *SALES PERSON-NUMBER* dan semua atribut yang tidak berulang (*SALES PERSON-NAME* dan *SALES-AREA*). Relasi kedua, *SALES PERSON-CUSTOMER*, berisi kunci utama dari relasi *SALES PERSON* (kunci utama *SALES PERSON* adalah *SALES PERSON-NUMBER*), serta semua atribut yang merupakan bagian dari grup berulang (*NOMOR PELANGGAN*, *NAMA PELANGGAN*, *GUDANG-NOMOR*, *GUDANG-LOKASI*, dan *SALESAMOUNT*). Mengetahui *NOMOR PENJUALAN*, bagaimanapun, tidak secara otomatis berarti bahwa Anda akan mengetahui *NAMA PELANGGAN*, *JUMLAH-PENJUALAN*, *LOKASI-GUDANG*, dan sebagainya. Dalam hubungan ini, seseorang harus menggunakan kunci bersambung (*SALES*

PERSON-NUMBER dan *CUSTOMER-NUMBER*) untuk mengakses informasi lainnya.

SALES PERSON (NOMOR PENJUALAN, NAMA PENJUALAN, AREA PENJUALAN) dan *SALESPERSON-CUSTOMER* (NOMOR PENJUALAN, NOMOR PELANGGAN, NAMA PELANGGAN, NOMOR GUDANG, LOKASI GUDANG, JUMLAH PENJUALAN)

Relasi *SALES PERSON-CUSTOMER* adalah relasi normal pertama, tetapi tidak dalam bentuk idealnya. Masalah muncul karena beberapa atribut tidak bergantung secara fungsional pada kunci primer [6].

2. Bentuk Normalisasi Kedua (2NF).



Gambar 2. 11 Contoh Bentuk Normalisasi Kedua (2NF)

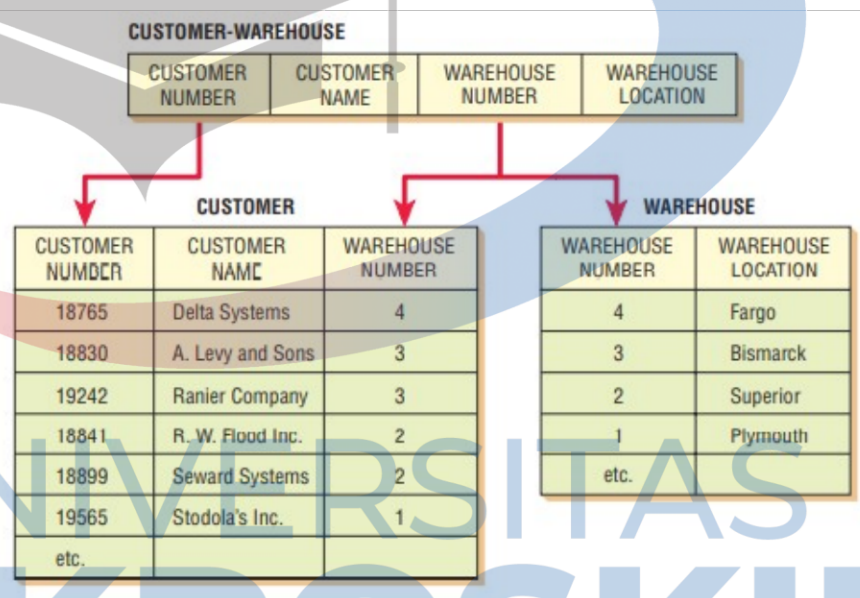
Dalam bentuk normalisasi kedua, semua atribut akan secara fungsional bergantung pada kunci utama. Oleh karena itu, langkah selanjutnya adalah menghapus semua atribut yang bergantung sebagian dan menempatkannya di relasi lain. Gambar diatas menunjukkan bagaimana relasi *SALES PERSON CUSTOMER* dibagi menjadi dua

relasi baru: *SALES* dan *CUSTOMER-WAREHOUSE*. Hubungan ini juga dapat diungkapkan sebagai berikut:

PENJUALAN (NOMOR PENJUALAN, NOMOR PELANGGAN, JUMLAH PENJUALAN) dan GUDANG PELANGGAN (NOMOR PELANGGAN, NAMA PELANGGAN, NOMOR GUDANG, LOKASI GUDANG).

Relasi PELANGGAN-GUDANG ada dalam bentuk normal kedua. Ini masih dapat disederhanakan lebih lanjut karena ada dependensi tambahan dalam relasinya. Beberapa atribut nonkey bergantung tidak hanya pada *primary key*, tetapi juga pada atribut *nonkey*. Ketergantungan ini disebut sebagai ketergantungan transitif [6].

3. Bentuk Normalisasi Ketiga (3NF)



Gambar 2. 12 Contoh Bentuk Normalisasi Ketiga

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

Kedua relasi baru tersebut disebut *customer* dan gudang, dan dapat ditulis sebagai berikut:

PELANGGAN (NOMOR PELANGGAN, NAMA PELANGGAN, NOMOR GUDANG) dan GUDANG (NOMOR GUDANG, LOKASI GUDANG)

Kunci utama untuk relasi *CUSTOMER* adalah *NOMOR PELANGGAN*, dan kunci utama untuk relasi *GUDANG* adalah *NOMOR GUDANG* [6].

2.4.5 Kamus Data

Kamus data adalah aplikasi khusus dari jenis kamus yang digunakan sebagai referensi dalam kehidupan sehari-hari. Kamus data adalah karya referensi data tentang data (yaitu, *meta data*), yang disusun oleh analis sistem untuk memandu mereka melalui analisis dan desain. Sebagai dokumen, kamus data mengumpulkan dan mengoordinasikan istilah data tertentu, dan menegaskan arti setiap istilah bagi orang yang berbeda dalam organisasi [6].

Selain menyediakan dokumentasi dan menghilangkan redundansi, kamus data dapat digunakan untuk [6]:

1. Memvalidasi diagram aliran data untuk kelengkapan dan keakuratannya.
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.
5. Membuat XML (*extensible markup language*).

Struktur data biasanya dijelaskan menggunakan notasi aljabar. Metode ini memungkinkan analis untuk menghasilkan tampilan dari elemen-elemen yang menyusun struktur data beserta informasi tentang elemen-elemen tersebut. Misalnya, analis 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 [6]:

1. Tanda sama dengan (=) berarti "terdiri dari".
2. Tanda plus (+) berarti "dan".
3. Tanda kurung {} 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. Tanda kurung [] mewakili situasi salah satu / atau. Salah satu elemen mungkin ada atau yang lain, tetapi tidak keduanya. Elemen yang tercantum di antara tanda kurung saling eksklusif.
5. Tanda kurung () mewakili elemen opsional. Elemen opsional dapat dikosongkan di layar entri dan mungkin berisi spasi atau nol untuk bidang numerik dalam struktur *file*.

Berikut ini contoh dari kamus data [6]:



Gambar 2. 13 Contoh Kamus Data

2.4.6 Structured English

Ketika logika proses melibatkan rumus atau iterasi, atau ketika keputusan terstruktur tidak kompleks, teknik yang tepat untuk menganalisis proses keputusan adalah penggunaan *Structured English*. Sesuai dengan namanya, *Structured English* didasarkan pada logika terstruktur, atau instruksi yang disusun dalam prosedur yang dikelompokkan dan pernyataan *Structured English* sederhana seperti menambah, mengalikan, dan memindahkan. Masalah kata dapat diubah menjadi bahasa Inggris

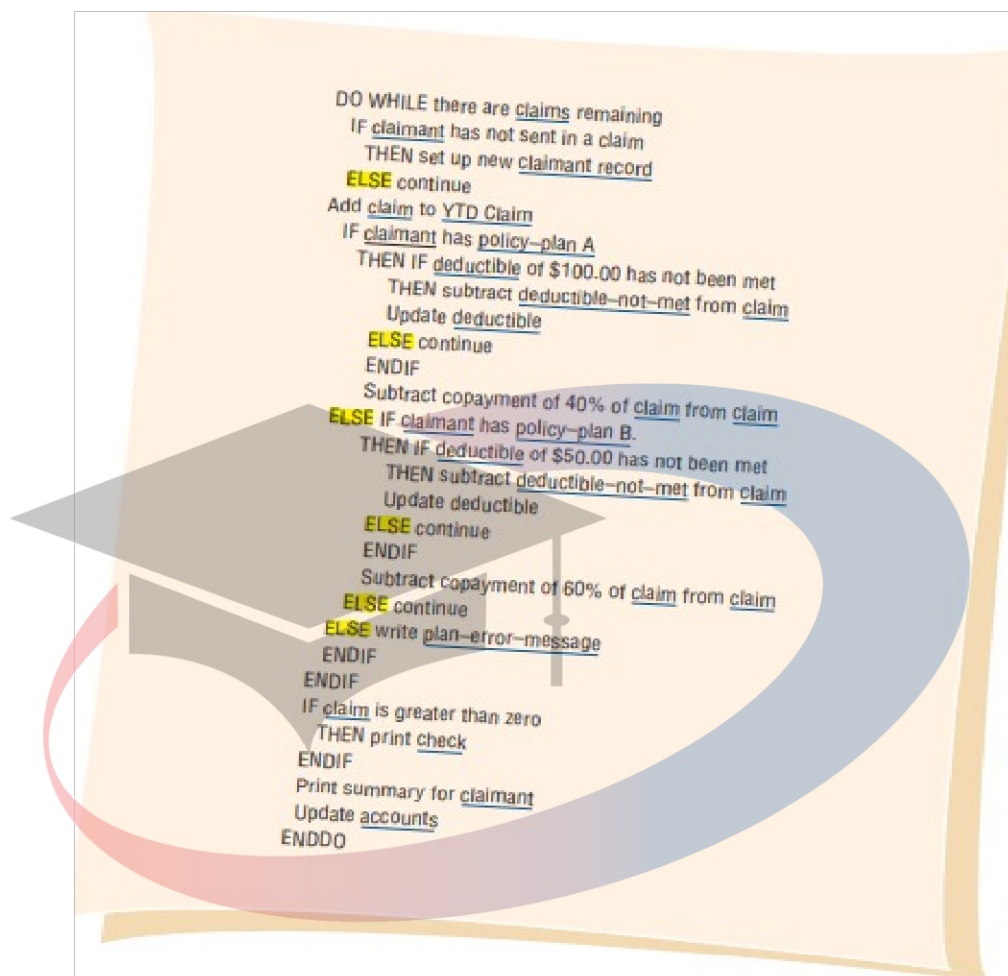
terstruktur dengan menempatkan aturan keputusan ke dalam urutan yang tepat dan menggunakan konvensi pernyataan *IF-THEN-ELSE* di seluruhnya [6].

Untuk menulis *Structured English*, mungkin ingin menggunakan konvensi sebagai berikut [6]:

1. Mengekspresikan semua logika dalam salah satu dari empat jenis berikut yaitu struktur sekuensial, struktur keputusan, struktur kasus, atau iterasihuruf.
2. Menggunakan kata kunci yang diterima seperti kapital seperti *IF*, *THEN*, *ELSE*, *DO*, *DO WHILE*, *DO*, dan *PERFORM*.
3. *Indent blok* pernyataan untuk menunjukkan hierarkinya (bersarang) dengan jelas.
4. Bila kata atau frasa telah didefinisikan dalam kamus data, garis bawah kata atau frasa tersebut untuk menandakan bahwa kata atau frasa tersebut memiliki makna khusus.
5. Berhati-hatilah saat menggunakan "dan" dan "atau," dan hindari kebingungan saat membedakan antara "lebih besar dari" dan "lebih besar dari atau sama dengan" dan hubungan sejenis. "A dan B" berarti A dan B; "A atau B" berarti A atau B, tetapi tidak keduanya. Klarifikasi pernyataan logis sekarang daripada menunggu sampai tahap pengkodean program.

Berikut contoh ini *Structured English* sebagai berikut [6]:

UNIVERSITAS
MIKROSKIL



Gambar 2. 14 Contoh Structured English

2.5 Basis Data

Database bukan hanya kumpulan *file*. Sebaliknya, *database* adalah sumber data pusat yang dimaksudkan untuk dibagikan oleh banyak pengguna untuk berbagai aplikasi. Inti dari basis data adalah sistem manajemen basis data (*DBMS*), yang memungkinkan pembuatan, modifikasi, dan pembaruan basis data; pengambilan data; dan pembuatan laporan dan tampilan. Orang yang memastikan bahwa *database* memenuhi tujuannya disebut administrator *database* [6].

Tujuan efektivitas *database* meliputi [6] :

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. Memungkinkan *database* berkembang seiring kebutuhan pengguna tumbuh.
5. Memungkinkan pengguna untuk membuat pandangan pribadi mereka terhadap data tanpa memperhatikan cara data disimpan secara fisik.

Daftar tujuan di atas memberi kita pengingat tentang keuntungan dan kerugian dari pendekatan *database*. Pertama, berbagi data berarti bahwa data hanya perlu disimpan satu kali. Itu pada gilirannya membantu mencapai integritas data, karena perubahan pada data dilakukan dengan lebih mudah dan andal jika data muncul sekali daripada di banyak *file* yang berbeda [6].

Ketika seorang pengguna membutuhkan data tertentu, *database* 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 di *database* daripada di sistem *file* konvensional. *Database* yang dirancang dengan baik juga bisa lebih fleksibel daripada *file* terpisah; artinya, basis data dapat berkembang seiring dengan perubahan kebutuhan pengguna dan aplikasi [6].

Bahasa basis data dapat dibedakan menjadi dua, yaitu [11] :

1. *Data Definition Language* (DDL)

DDL merupakan kumpulan dari *Structured Query Language* (SQL) yang digunakan untuk membuat (*create*), mengubah (*alter*), dan menghapus (*drop*) struktur dan definisi tipe data dari objek-objek *database*. Dengan bahasa ini dapat membuat tabel baru, membuat *indeks*, mengubah tabel, menentukan struktur tabel, dan lain-lain. Hasil dari kompilasi perintah DDL menjadi kamus data, yaitu data yang menjelaskan data sesungguhnya.

2. *Data Manipulation Language* (DML)

DML merupakan kumpulan perintah *query* yang digunakan untuk memanipulasi data pada *database*. DML berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data, yang berupa *insert*, *update*, *delete*, dan lain-lain. Ada dua jenis, yaitu *procedural* (ditentukan data yang diinginkan dan cara mendapatkannya) dan *non-prosedural* (tanpa menyebutkan cara mendapatkannya).

2.6 Penjualan

Penjualan merupakan sebuah proses dimana kebutuhan pembeli dan kebutuhan penjual dipenuhi melalui antar pertukaran informasi dan kepentingan. Jadi konsep penjualan adalah cara untuk mempengaruhi konsumen untuk membeli produk yang ditawarkan atau dijual [12].

Dalam kenyataannya penjualan mempunyai dua sistem yang biasa diterapkan oleh baik suatu usaha atau orang yaitu penjualan yang dilakukan dengan cara tunai dan penjualan yang dilakukan secara kredit atau sering disebut angsuran. Penjualan yang dilakukan secara tunai merupakan penjualan dimana terjadi penjualan, pembeli akan membayar harga barang atau jasa yang dibelinya saat itu juga. Penjualan yang dilakukan secara kredit atau angsuran adalah pembayaran baru diterima beberapa waktu kemudian setelah terjadinya transaksi penjualan dan cara pembayaran dapat dilakukan dengan jumlah tertentu dan dalam jangka waktu tertentu [12].

Penjualan merupakan strategi utama perusahaan untuk dapat menjalankan suatu perusahaan. Penjualan yang terjadi dapat menimbulkan laba pada pihak yang menjualnya. Ketika perusahaan mengalami banyak penjualan maka perusahaan akan menerima laba. Semakin tinggi angka penjualan maka semakin tinggi juga laba yang diterima. Sebaliknya sedikit penjualan yang terjadi maka sedikit juga laba yang diterima [13].

Perusahaan selalu mempunyai banyak cara untuk meningkatkan penjualan. Ketika penjualan berkurang pasti perusahaan akan dapat mengalami kerugian. Karena dengan laba dari penjualan yang sedikit tidak akan dapat menghidupkan atau menjalankan perusahaan tersebut dengan lancar. Penjualan dikenal dengan dua cara, yaitu [13]:

1. Penjualan Tunai merupakan pembayaran barang/jasa yang dibayar dengan lunas pada saat pengambilan barang/jasa tersebut.
2. Penjualan Kredit adalah penjualan dengan metode pembayaran berangsur. Dengan membayar uang muka dan uang setoran bulanan pembeli dapat membawa pulang barang/jasa yang dibeli. Setiap pengajuan pembelian kredit ada syarat-syarat yang diberikan oleh pihak perusahaan. Ketika proses penjualan

kredit terjadi akan timbul piutang, yaitu perusahaan akan menagih uang tagihan sesuai waktu dan nominal yang disepakati.

2.6.1 Piutang

Piutang merupakan bentuk penjualan yang dilakukan oleh suatu perusahaan dimana pembayarannya tidak dilakukan secara tunai, namun bersifat bertahap. Penjualan piutang artinya lebih jauh perusahaan menerapkan manajemen kredit. Dan salah satu target dari manajemen kredit adalah tercapainya target penjualan sesuai dengan perencanaan, serta selanjutnya menunggu masuknya dana angsuran ke kas perusahaan [14].

Piutang adalah klaim atas uang, barang, atau jasa kepada pelanggan atau pihak-pihak lainnya. Piutang usaha pada umumnya adalah kategori yang paling signifikan dari piutang dan merupakan hasil dari aktivitas normal perusahaan atau entitas, yaitu penjualan barang atau jasa secara kredit kepada pelanggan. Piutang usaha dapat diperkuat dengan janji pembayaran tertulis secara formal dan diklasifikasikan sebagai wesel tagih (*notes receivable*). Piutang usaha umumnya merupakan jumlah yang material di neraca bila dibandingkan dengan piutang non-usaha [14].

Piutang dagang (*trade receivable*) dihasilkan dari kegiatan normal bisnis perusahaan yaitu penjualan secara kredit barang atau jasa ke pelanggan. Piutang dagang yang di buktikan dengan sebuah janji tertulis secara formal oleh pelanggan untuk membayar, diklasifikasikan sebagai piutang wesel (*notes receivable*). Dalam kebanyakan kasus akan tetapi, piutang dagang merupakan piutang kepada pelanggan yang tanpa adanya jaminan dari pelanggan untuk membayar atau “*open accounts*”, yang sering dikenal sebagai piutang usaha (*notntrade receivable*) meliputi seluruh jenis piutang lainnya seperti yang telah disebutkan di atas, yaitu piutang bunga, piutang deviden, piutang pajak, tagihan kepada perusahaan asosiasi, dan tagihan kepada karyawan [15].

Ada dua jenis piutang usaha/ piutang dagang yang perlu kita ketahui, antara lain piutang dagang, piutang wesel, dan piutang lain-lain. Untuk lebih jelasnya tentang 3 jenis piutang tersebut, berikut ini penjelasannya [15] :

1. Piutang dagang (*accounts receivable*)

Jenis piutang yang pertama yaitu piutang dagang. Piutang dagang adalah tagihan perusahaan kepada pelanggan sebagai akibat adanya penjualan barang atau jasa secara kredit. Jenis piutang ini sangat sering kita jumpai dalam transaksi keuangan. Piutang dagang umumnya berjangka waktu kurang dari satu tahun sehingga dilaporkan sebagai aktiva lancar. Untuk meningkatkan motivasi para customer kita dalam membayar piutang, biasanya dalam transaksi penjualan kredit disertai adanya potongan piutang jika tepat waktu, dan adanya denda jika terjadi keterlambatan dalam pembayaran piutang oleh pelanggan.

2. Piutang wesel (*notes receivable*)

Jenis piutang wesel yaitu piutang berupa perjanjian tertulis dari debitur kepada kreditur untuk membayar sejumlah uang yang tercantum dalam surat perjanjian tersebut pada waktu tertentu di masa yang akan datang. Umumnya jangka wesel berjangka tersebut lebih dari 60 hari, apabila piutang wesel berjangka waktu kurang dari satu tahun dilaporkan dalam neraca sebagai aktiva lancar sedangkan untuk piutang wesel waktu lebih dari satu tahun diperlukan sebagai piutang jangka panjang.

3. Piutang Non Dagang

Piutang non dagang adalah tagihan perusahaan kepada pihak lain atau pihak ketiga yang timbul atau terjadi bukan karena adanya transaksi penjualan barang dagang atau jasa secara kredit. Jumlah piutang non dagang/lainnya biasanya tidak signifikan dibandingkan dengan jumlah piutang dagang ataupun piutang usaha.