

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

Pada bagian ini, penulis akan menjelaskan tentang bagaimana konsep dari sistem informasi yang akan dijelaskan berikut ini.

2.1.1 Sistem

Sistem merupakan kumpulan dari elemen-elemen yang saling berhubungan yang bekerja sama untuk memproses masukan yang diarahkan pada sistem (*input*) dan mengolah masukan tersebut untuk menghasilkan keluaran yang diinginkan. Sistem yang baik harus memiliki tujuan yang tepat, karena hal ini sangat penting dalam menentukan *input* yang dibutuhkan sistem dan *output* yang dihasilkan [1]. Sistem terdiri dari seperangkat elemen atau komponen yang berinteraksi untuk mencapai tujuan. Elemen itu sendiri dan hubungannya menentukan bagaimana sistem bekerja. Sistem memiliki *input*, mekanisme pemrosesan, *output* dan umpan balik. suatu sistem memproses *input* untuk menghasilkan *output* [2].

Tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Sistem ada karena tujuan. Sistem dibangun agar tujuan tercapai tidak menyimpang sehingga risiko kegagalan bisa diminimalkan. Agar supaya target tersebut bisa tercapai secara efektif dan efisien maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya agar sistem dapat dibangun dan menuntun dengan jelas dan tegas setiap aktivitas menuju tujuan yang telah ditetapkan [3].

Sistem terdiri dari 3 fungsi, yang menunjukkan bahwa sistem sebagai proses tidak bisa berdiri sendiri, yaitu : [3]

1. *Input*

Input adalah segala sesuatu yang masuk kedalam suatu sistem. *Input* ini bervariasi bisa berupa energi, manusia, data, modal, bahan baku, layanan atau lainnya. *Input* merupakan pemicu bagi sistem untuk melakukan proses yang diperlukan.

2. *Process*

Proses merupakan perubahan dari *input* menjadi *output*. Proses mungkin dilakukan oleh mesin, orang, atau komputer. Umumnya kita mengetahui bagaimana *input* dirubah menjadi *output* akan tetapi pada situasi tertentu proses tidak diketahui secara detail karena perubahan ini terlalu kompleks. Proses mungkin berupa perakitan yang

menghasilkan satu macam *output* dari berbagai macam *input* yang disusun berdasarkan aturan tertentu.

3. *Output*

Output seperti halnya input mungkin berbentuk produk, servis, informasi dalam bentuk *print out* komputer atau energi seperti *output* dari dinamo. *Output* adalah hasil dari suatu proses yang merupakan tujuan dari keberadaan sistem. Seperti dijelaskan sebelumnya *output* suatu sistem bisa menjadi input untuk sistem yang lain yang setelah diproses menjadi *output* yang lain.

Model umum sebuah sistem terdiri dari *input*, proses dan *output*. Hal ini merupakan sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus, selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem, Adapun karakteristik yang dimaksud, yaitu: [4]

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama 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 apa pun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara.

4. Penghubung Sistem (*Interface*)

Media penghubung merupakan 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.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer sementara “data” adalah signal input yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi sub-sistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, yang mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi sub-sistem lainnya.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran, maka sistem operasi tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.2 Informasi

Informasi adalah data yang telah diklasifikasikan atau di olah ataupun diinterpretasikan untuk digunakan dalam sistem proses pengambilan keputusan. Sistem pengolahan sistem informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan [4].

Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu yang nyata, dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi. Perubahan data informasi dilakukan oleh pengolah informasi (*Information processor*). Pengolah informasi merupakan salah satu elemen kunci dalam

sistem konseptual. Pengolah informasi dapat meliputi elemen-elemen komputer, elemen non-komputer atau kombinasi [4].

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, yaitu [4]:

1. Informasi Strategis. Informasi ini digunakan untuk mengambil keputusan jangka panjang yang mencakup informasi eksternal, rencana perluasan perusahaan, dsb.
2. Informasi Taktis. Informasi ini dibutuhkan untuk mengambil keputusan menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi Teknis. Informasi ini dibutuhkan untuk keperluan operasional sehari-hari seperti informasi persediaan stok, retur penjualan dan laporan khas harian.

Ada 3 hal yang menjadi penentu kualitas sebuah informasi, yaitu informasi yang harus akurat, tepat waktu dan relevan. Berikut penjelasan dari ketiga hal tersebut:

1. Akurat (*Accurate*)

Informasi harus terukur, jelas, dan bebas dari kesalahan, karena mungkin ada banyak gangguan dari sumber informasi ke penerima informasi, sehingga informasi tidak dapat diterima secara keseluruhan.

2. Tepat waktu (*Timeliness*)

Informasi harus sampai ke penerima tepat waktu. Karena jika informasi tidak sampai tepat waktu, bisa jadi informasi tersebut sudah tidak berguna lagi bagi penerimanya karena akan digunakan sebagai dasar pengambilan kesimpulan. Jika informasinya tidak jelas, maka akan berdampak serius bagi perusahaan. Saat ini, informasi sangat penting dan mahal karena kecepatan pertukaran informasi, baik dikirim atau tidak, sangat penting. Hal ini membutuhkan teknologi canggih untuk menemukan, memproses dan mengirimkan informasi ini.

3. Relevan (*Relevance*)

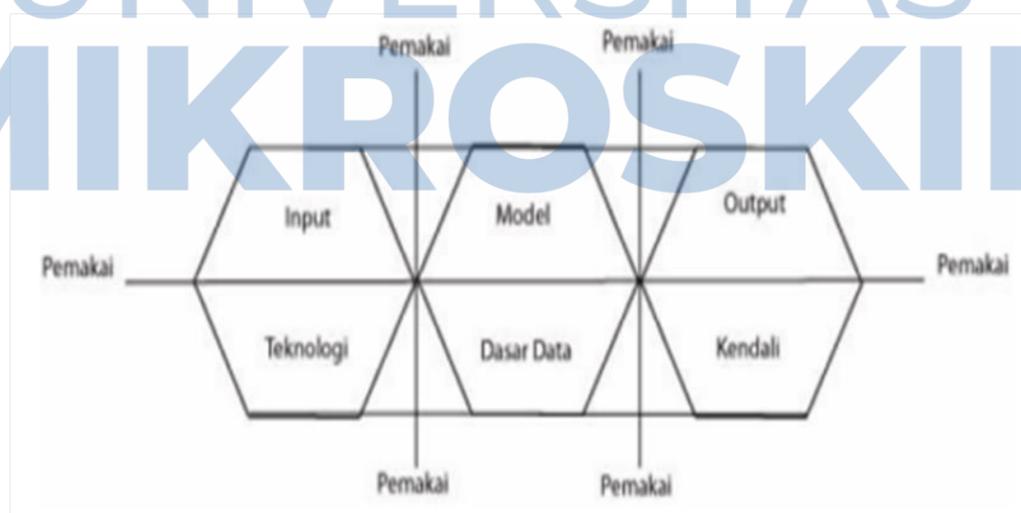
Informasi yang ada harus bermanfaat bagi penggunanya. Relevansi informasi berbeda untuk satu orang dan orang lain. Misalnya, informasi tentang penyebab kegagalan mesin tidak relevan jika dikirim ke departemen pemasaran. Lebih penting lagi jika kerusakan mesin diperbaiki oleh mekanik perusahaan. Begitu pula sebaliknya, informasi tentang departemen keuangan perusahaan lebih relevan jika dikirim ke departemen akuntansi perusahaan daripada departemen mekanik perusahaan.

Informasi ditentukan oleh dua hal yaitu manfaat dan biaya perolehannya. Namun, informasi lebih berharga jika memiliki manfaat dibandingkan dengan biaya untuk memperolehnya [5].

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 laporan-laporan yang diperlukan oleh pihak luar tertentu. Sistem informasi bukan merupakan hal yang baru. Hal yang dimaksud baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi memungkinkan manajer merencanakan serta mengendalikan operasi telah ada. Komputer menambah satu atau dua dimensi, seperti kecepatan, ketelitian dan penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan [4].

Sistem Informasi terdiri dari komponen-komponen yang disebut blok bangunan yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing harus saling berinteraksi dengan yang lain untuk membentuk suatu kesatuan untuk mencapai sasaran [4].



Gambar 2. 1 Komposisi Sistem Informasi

Berikut ini sistem informasi terbagi menjadi 6 blok, sebagai berikut [4]:

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, 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 tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

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

4. Blok Teknologi (*Technology Block*)

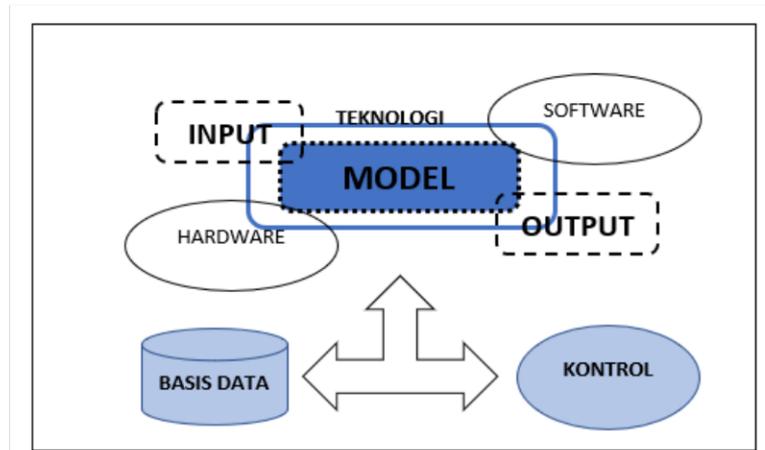
Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain. Data yang di simpan dalam *database* untuk keperluan penyediaan informasi lebih lanjut.

6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi. Maka dari itu beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal tersebut dapat dicegah dan jika sudah terjadi dapat langsung diatasi [4].



Gambar 2.2 Komposisi Komponen Sistem Informasi

2.2. System Development Life Cycle (SDLC)

Data Flow Diagram (DFD) adalah sebuah gambaran dari sistem yang sudah ada atau sistem baru yang dikembangkan dengan logika tanpa mempertimbangkan lingkungan fisik di mana data tersebut berjalan. Dengan adanya DFD maka pengguna sistem yang kurang paham dalam bidang komputer dapat mengerti sistem yang sedang berjalan. Di dalam DFD terdapat 3 level yaitu: [6]

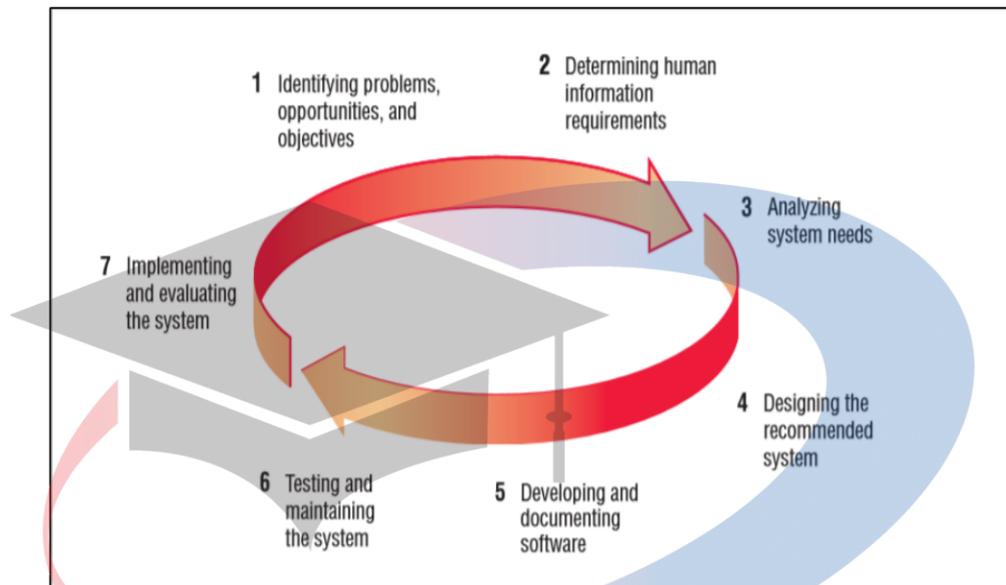
1. Diagram Konteks

Menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Diagram konteks merupakan tingkatan tertinggi dalam DFD dan biasanya diberi nomor 0. Semua entitas eksternal yang ditunjukkan pada diagram konteks aliran-aliran data utama menuju dan dari sistem. Fungsi dari diagram konteks adalah untuk membatasi sistem dan menunjukkan adanya interaksi sistem dengan komponen di luar sistem, maka perlu dibuat diagram konteks yang merupakan gambaran sistem secara keseluruhan. Diagram konteks ini menggambarkan aliran pengolahan data keanggotaan, penjualan barang dan pembelian barang serta informasi dari sistem informasi pengolahan data penjualan dan pembelian barang. Dalam diagram konteks tidak ada data store karena diagram konteks hanya menggambarkan hubungan antara sistem dengan entitas luar. Secara umum berikut aturan dalam diagram konteks :

- a. Menggambarkan hubungan sistem dengan entitas luar yang berhubungan langsung dengan sistem.
- b. Terdapat komponen sistem, entitas, aliran data.
- c. Aliran data dari entitas ke sistem maupun sebaliknya.

- d. Tidak boleh dituliskan aliran data dari entitas ke entitas langsung tanpa melalui sistem, jika memang hal itu terjadi dalam proses yang ada maka tidak perlu digambarkan atau dituliskan pada diagram konteks.

Disini kita telah membagi siklus menjadi tujuh fase, seperti yang ditunjukkan pada gambar berikut ini :



Gambar 2.3 *System Development Life Cycle (SDLC)*

Berikut masing-masing penjelasan dari *System Development Life Cycle* [7]:

1. Mengidentifikasi masalah, peluang, dan tujuan

Dalam fase pertama siklus hidup pengembangan sistem ini, analisis berkepentingan untuk mengidentifikasi dengan benar masalah, peluang, dan tujuan. Tahap ini sangat penting untuk keberhasilan proyek, karena tidak ada yang mau menyalahkan waktu berikutnya untuk mengatasi masalah yang salah.

Fase pertama mengharuskan analisis untuk melihat dengan jujur apa yang terjadi dalam bisnis. Kemudian, bersama dengan anggota organisasi lainnya, analisis menunjukkan masalah. Sering kali orang lain akan memunculkan masalah ini, dan mereka adalah alasan analisis awalnya dipanggil. Peluang adalah situasi yang diyakini analisis dapat ditingkatkan melalui penggunaan sistem informasi yang terkomputerisasi. Memanfaatkan peluang dapat memungkinkan bisnis untuk mendapatkan keunggulan kompetitif atau menetapkan standar industri.

Identifikasi tujuan juga merupakan komponen penting dari fase pertama. Analisis pertama-tama harus menemukan apa yang coba dilakukan oleh bisnis. Kemudian analisis

akan dapat melihat apakah beberapa aspek aplikasi sistem informasi dapat membantu bisnis mencapai tujuannya dengan mengatasi masalah atau peluang tertentu.

Orang-orang yang terlibat dalam fase pertama adalah para pengguna, analis, dan manajer sistem yang mengkoordinasi proyek. Kegiatan dalam fase ini terdiri dari wawancara manajemen pengguna, meringkas pengetahuan yang diperoleh, memperkirakan ruang lingkup proyek, dan mendokumentasikan hasilnya. Output dari fase ini adalah laporan kelayakan yang berisi definisi masalah dan meringkas tujuan. Manajemen kemudian harus membuat keputusan apakah akan melanjutkan proyek yang diusulkan. Jika pengguna tidak memiliki dana yang cukup dalam anggarannya atau ingin mengatasi masalah yang tidak terkait atau jika masalah tidak memerlukan sistem komputer, solusi yang berbeda mungkin disarankan, dan proyek sistem tidak dilanjutkan lebih jauh.

2. Menentukan syarat-syarat informasi

Fase selanjutnya yang dimasukkan analis adalah menentukan kebutuhan pengguna yang terlibat, menggunakan berbagai alat untuk memahami bagaimana pengguna berinteraksi dalam konteks kerja dengan sistem informasi mereka saat ini. Analis akan menggunakan metode interaktif seperti wawancara, pengambilan sample dan investigasi data keras, dan kuesioner, bersama dengan metode yang tidak mengganggu, seperti mengamati perilaku pengambil keputusan dan lingkungan kantor mereka, dan metode yang mencakup semua, seperti *prototyping*.

Analis akan menggunakan metode ini untuk mengajukan dan menjawab banyak pertanyaan tentang *human computer interaction* (HCI), termasuk pertanyaan seperti, “Apa kekuatan dan keterbatasan fisik pengguna?” Dengan kata lain, “Apa yang perlu dilakukan untuk membuat sistem terdengar, dapat dibaca dan aman?” “Bagaimana sistem yang baru dirancang agar mudah digunakan, dipelajari dan diingat?” “Bagaimana sistem bisa dibuat menyenangkan atau bahkan menyenangkan untuk digunakan?” “Bagaimana sistem dapat mendukung suatu tugas pekerjaan individu dan membuatnya lebih produktif dengan cara baru?”.

Dalam informasi persyaratan fase SDLC, analisis ini berusaha untuk mengetahui apa yang pengguna butuhkan untuk melakukan pekerjaan mereka. Pada titik ini analis sedang memeriksa bagaimana membuat sistem berguna bagi orang-orang yang terlibat. Bagaimana sistem dapat lebih baik mendukung tugas individu yang perlu dilakukan? Apa tugas baru yang diaktifkan oleh sistem baru yang dapat dilakukan pengguna? Bagaimana sistem baru dapat dibuat untuk memperluas kemampuan pengguna di luar

apa yang disediakan sistem lama? Bagaimana analis dapat membuat sistem yang bermanfaat bagi pekerja untuk digunakan?

Orang-orang yang terlibat dalam fase ini adalah analis dan pengguna, biasanya manajer operasi dan pekerja operasi. Analis sistem perlu mengetahui detail fungsi sistem saat ini: siapa (orang-orang yang terlibat), apa (aktivitas bisnis), dimana (lingkungan pekerjaan berlangsung), kapan (waktu), dan bagaimana (bagaimana prosedur saat ini dilakukan) dari bisnis yang diteliti. Analis kemudian harus bertanya mengapa bisnis menggunakan sistem saat ini. Mungkin ada alasan bagus untuk melakukan bisnis menggunakan metode saat ini, dan ini harus dipertimbangkan ketika merancang sistem terbaru.

3. Menganalisis kebutuhan sistem

Fase berikutnya yang dilakukan oleh analis sistem melibatkan analisis kebutuhan sistem. Sekali lagi, alat dan teknik khusus membantu analis membuat penentuan kebutuhan. Alat-alat seperti *Data Flow Diagram* (DFD) untuk memetakan *input*, proses, dan *output* fungsi bisnis, atau diagram aktivitas atau diagram urutan untuk menunjukkan urutan peristiwa, menggambarkan sistem dalam bentuk grafis terstruktur. Dari aliran data, urutan, atau diagram lain, kamus data dikembangkan yang mencantumkan semua item data yang digunakan dalam sistem, serta spesifikasinya.

Selama fase ini analis sistem juga menganalisis keputusan terstruktur yang dibuat. Keputusan terstruktur adalah keputusan yang menentukan kondisi, alternatif kondisi, tindakan, dan aturan tindakan. Ada tiga metode utama untuk analisis keputusan terstruktur: bahasa Inggris terstruktur, tabel keputusan, dan pohon keputusan.

Pada titik ini di SDLC, analis sistem menyiapkan proposal sistem yang merangkum apa yang telah diketahui tentang pengguna, kegunaan, 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, analis melanjutkannya. Setiap masalah sistem adalah unik, dan tidak pernah ada satu solusi yang benar. Cara di mana rekomendasi atau solusi dirumuskan tergantung pada kualitas individu dan pelatihan profesional masing-masing analis dan interaksi analis dengan pengguna dalam konteks lingkungan kerja mereka.

4. Merancang sistem yang direkomendasikan

Dalam fase desain SDLC, analis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk mencapai desain logis dari sistem informasi. Analis merancang prosedur bagi pengguna untuk membantu mereka memasukkan data secara akurat

sehingga data yang masuk ke sistem informasi benar. Selain itu, analis menyediakan bagi pengguna untuk menyelesaikan input efektif ke sistem informasi dengan menggunakan teknik bentuk yang baik dan halaman Web atau desain layar.

Bagian dari desain logis dari sistem informasi adalah merancang HCI. Antarmuka menghubungkan pengguna dengan sistem dan dengan demikian sangat penting. Antarmuka pengguna dirancang dengan bantuan pengguna untuk memastikan bahwa sistemnya dapat didengar, terbaca, dan aman, serta menarik dan menyenangkan untuk digunakan. Contoh antarmuka pengguna fisik termasuk *keyboard* (untuk mengetik pertanyaan dan jawaban), menu pada layar (untuk memperoleh perintah pengguna), dan berbagai antarmuka pengguna grafis (GUI) yang menggunakan mouse atau layar sentuh. Fase desain juga mencakup perancangan basis data yang akan menyimpan banyak data yang dibutuhkan oleh pembuat keputusan dalam organisasi. Pengguna mendapat manfaat dari *database* yang terorganisir dengan baik yang logis bagi mereka dan sesuai dengan cara mereka melihat pekerjaan mereka. Dalam fase ini analis juga bekerja dengan pengguna untuk mendesain output (baik pada layar atau dicetak) yang memenuhi kebutuhan informasi mereka.

Akhirnya, analis harus merancang kontrol dan prosedur cadangan untuk melindungi sistem dan data, dan untuk menghasilkan paket spesifikasi program untuk programmer. 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 alir data, dan nama dan fungsi dari setiap kode yang ditulis sebelumnya baik yang ditulis sendiri atau menggunakan kode atau pustaka kelas lainnya.

5. Mengembangkan dan Mendokumentasikan Perangkat Lunak

Pada fase kelima SDLC, analis bekerja dengan programmer untuk mengembangkan perangkat lunak asli yang diperlukan. Selama fase ini analis bekerja dengan pengguna untuk mengembangkan dokumentasi yang efektif untuk perangkat lunak, termasuk manual prosedur, bantuan online, dan situs Web yang menampilkan *Frequently Asked Questions* (FAQ), pada *file Read Me* yang dikirimkan dengan perangkat lunak baru. Karena pengguna terlibat sejak awal, dokumentasi fase harus menjawab pertanyaan yang mereka ajukan dan selesaikan bersama dengan analis. Dokumentasi memberi tahu pengguna cara menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi masalah perangkat lunak.

Programmer mencukur peran kunci dalam fase ini karena mereka merancang, kode, dan menghapus kesalahan sintaksis dari program komputer. Untuk memastikan kualitas,

seorang *programmer* dapat melakukan desain atau kode berjalan, menjelaskan bagian-bagian kompleks dari program kepada tim *programmer* lain.

6. Menguji dan mempertahankan sistem

Sebelum sistem informasi dapat digunakan, itu harus diuji. Hal ini jauh lebih murah untuk menangkap masalah sebelum sistem masuk ke pengguna. Beberapa pengujian diselesaikan oleh pemrogram saja, sebagian lagi oleh analis sistem bersama dengan pemrogram. Pemeriksaan tes untuk menunjukkan masalah dijalankan pertama dengan data sampel dan akhirnya dengan data aktual dari sistem saat ini. Seringkali rencana pengujian dibuat di awal SDLC dan disempurnakan seiring proyek berlangsung.

Pemeliharaan sistem dan dokumentasinya dimulai pada fase 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 prosedur sistematis yang digunakan analis di seluruh SDLC dapat membantu memastikan bahwa pemeliharaan dijaga agar tetap minimum.

7. Mengimplementasikan dan mengevaluasi sistem.

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

Evaluasi dimasukkan sebagai bagian dari fase akhir SDLC ini sebagian besar demi diskusi. Sebenarnya, evaluasi berlangsung selama setiap fase. Kriteria utama yang harus dipenuhi adalah apakah pengguna yang dituju memang menggunakan sistem. Perlu dicatat bahwa kerja sistem sering kali bersifat siklis. Ketika seorang analis menyelesaikan satu fase pengembangan sistem dan melanjutkan ke tahap berikutnya, penemuan masalah dapat memaksa analis untuk kembali ke fase sebelumnya dan memodifikasi pekerjaan yang dilakukan di sana.

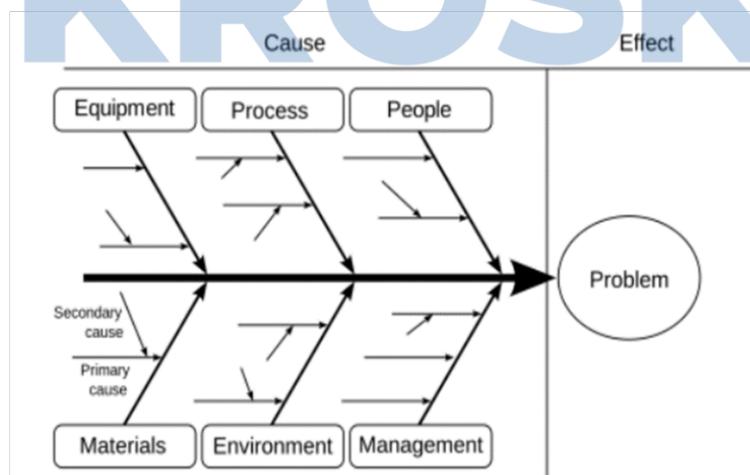
2.3 Alat Bantu Pengembangan Sistem

Pada bagian ini, akan dijelaskan mengenai alat bantu pengembangan system yang dipakai yang ada pada penjelasan di bawah ini.

2.3.1 Diagram Fishbone

Diagram *Fishbone* (Tulang Ikan), atau Diagram Ishikawa dikembangkan pertama kali oleh Kaoru Ishikawa. Nama “Tulang Ikan” sering digunakan karena bentuk diagram ini menyerupai tampak samping dari tulang ikan. Diagram ini digunakan untuk desain produk dan mencegah terjadinya *defect*, dengan menganalisis dan menetapkan faktor penyebab yang paling berpengaruh dalam terjadinya *defect*. Permasalahan yang akan diperbaiki diletakkan pada “Kepala Ikan”, dan setiap “Tulang Ikan” terbesar dalam diagram mewakili kategori penyebab utama. Secara umum kategori-kategori pada Diagram *Fishbone* terdiri atas hal-hal berikut [8] :

1. *Men/People*: sumber daya manusia yang terlibat dalam proses.
2. *Method*: bagaimana proses dilaksanakan dan persyaratan spesifik apa saja yang dibutuhkan untuk melaksanakan proses tersebut seperti kebijakan, prosedur dan peraturan perundangan.
3. *Machine*: seluruh peralatan, *computer*, perangkat yang dibutuhkan untuk melaksanakan proses.
4. *Material*: bahan mentah, bahan baku, suku cadang, alat tulis, dan bahan-bahan lainnya yang digunakan sebagai input proses untuk membuat produk akhir.
5. *Measurement*: data kuantitas/kualitas kinerja yang diperoleh dari proses yang digunakan untuk mengevaluasi mutu serta teknik yang digunakan untuk mengumpulkan data.
6. *Environment*: kondisi seperti lokasi, waktu, suhu, dan budaya dimana proses beroperasi.



Gambar 2.4 Contoh Diagram Fishbone

Untuk menetapkan penyebab terjadinya *defect* pada proses dapat menggunakan metode *brainstorming*, untuk kemudian dikelompokkan sesuai dengan kategori pada “Tulang Ikan”. Penyebab masalah yang lebih detail kemudian ditempatkan sebagai cabang dari “Tulang Ikan” terbesar, hingga ditemukan akar masalah. Tujuan dari analisis sebab akibat menggunakan Diagram Tulang Ikan sebagai berikut [8] :

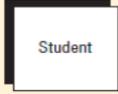
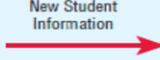
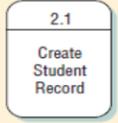
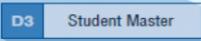
1. Untuk mengenali penyebab penting terjadinya *defect*
2. Untuk memahami semua akibat dan penyebab terjadinya *defect*
3. Untuk membandingkan prosedur kerja
4. Untuk menemukan pemecahan masalah yang tepat
5. Untuk mengidentifikasi hal apa yang harus dilakukan
6. Untuk mengembangkan proses

2.3.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan sebuah alat dokumentasi berbentuk grafik yang menjelaskan suatu proses dalam perusahaan menggunakan simbol-simbol, biasanya diagram konteks tersebut terdiri dari suatu proses dan menggambarkan suatu ruang lingkup dari sistem [9]. Dalam diagram DFD, diagram konteks merupakan level tertinggi yang biasanya menggambarkan seluruh *input* ke dalam sistem atau *output* dari sebuah sistem yang dibatasi oleh *boundary*. Diagram konteks meliputi beberapa sistem antara lain [9]:

1. Kelompok pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai *terminator*).
2. Data masuk, data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data keluar, data yang dihasilkan sistem dan diberikan ke dunia luar.
4. Penyimpanan data (*storage*), digunakan secara bersama antara sistem dengan terminator.
5. Batasan (*boundary*), antara sistem dengan lingkungan luar.

Dalam pembuatan DFD dapat digunakan dengan menggunakan simbol-simbol berikut :

Symbol	Meaning	Example
	Entity	
	Data Flow	
	Process	
	Data Store	

Gambar 2.5 Simbol-Simbol DFD

Terdapat empat macam simbol dalam *data flow diagram* (DFD) dalam menyatakan aliran proses seluruh sistem [9] :

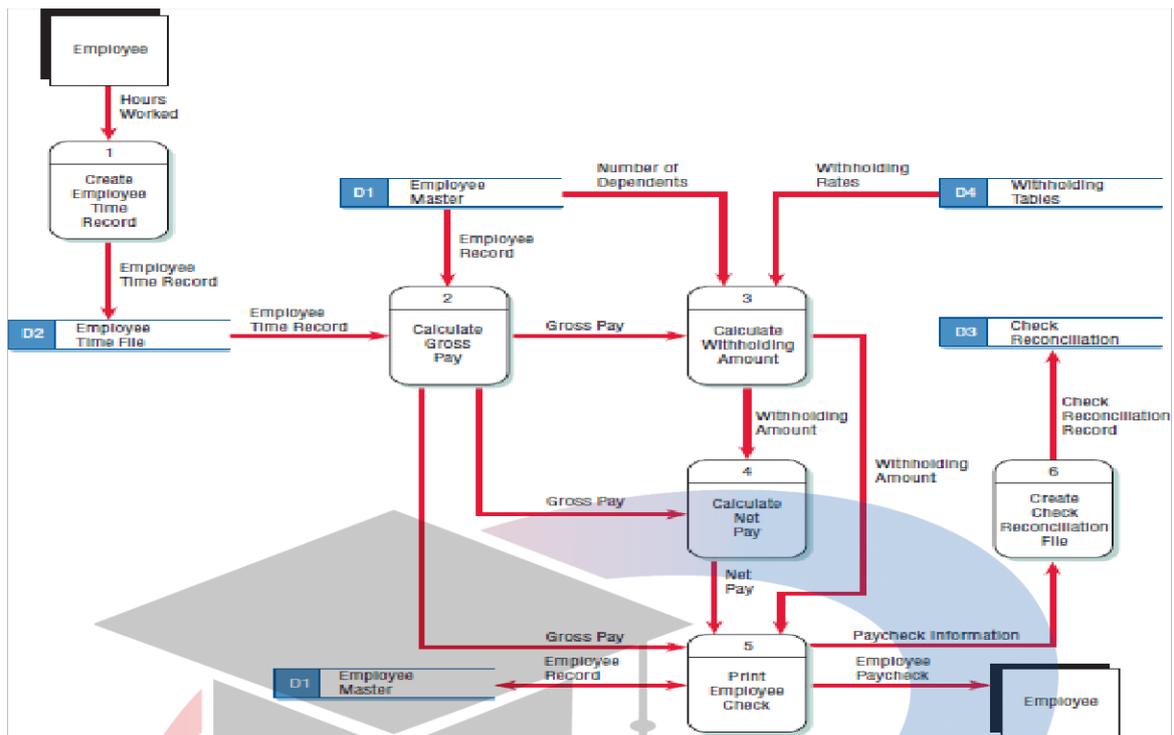
- External entity* (kesatuan luar) atau *boundary* (batas sistem), digunakan untuk menyatakan sumber atau tujuan dari aliran data suatu sistem. Simbol dari *external entity* dilambangkan dengan Persegi Panjang atau Kotak.
- Data flow* (arus data), digunakan untuk menunjukkan aliran data dari suatu proses ke proses lain, digambarkan simbol tanda panah
- Process* (proses), digunakan untuk menunjukkan suatu input yang ditransformasikan menjadi output dari proses bisnis. Biasanya proses digambarkan dalam bentuk lingkaran maupun oval.
- Data store* (simpanan data) digunakan untuk menunjukkan simpanan dari data yang dapat digunakan untuk proses selanjutnya, penggambaran simbol *data store* berupa dua garis sejajar.

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

- 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. Setiap proses mentransformasikan data dan harus menerima dan menghasilkan keluaran.

2. Menghubungkan penyimpanan data dan entitas-entitas eksternal secara langsung satu sama lain. Penyimpanan data serta entitas juga tidak perlu dikoneksikan satu sama lain. Penyimpanan data dan entitas eksternal hanya terhubung dengan suatu proses.
3. Aliran data-aliran data atau proses-proses pemberian label yang tidak tepat. Periksa diagram aliran data tersebut untuk memastikan bahwa setiap objek atau aliran data diberi label yang sesuai. Sebuah proses harus menunjukkan nama sistem atau menggunakan format kata kerja-kata sifat-kata benda. Masing-masing aliran data harus bisa digambarkan dengan sebuah kata benda.
4. Memasukkan lebih dari Sembilan proses pada diagram aliran data. Memiliki terlalu banyak proses yang menciptakan suatu diagram yang kacau akan memusingkan untuk dibaca dan malah menghalangi komunikasi. Bila melibatkan lebih dari Sembilan proses dalam suatu sistem, kelompokkan beberapa proses yang bekerja Bersama-sama didalam suatu subsistem dan letakkan mereka pada suatu diagram anak.
5. Mengabaikan aliran data. Perhatikan aliran linear dalam diagram, maksudnya, 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. Masing-masing diagram anak harus memiliki masukan dan aliran data keluaran yang sama seperti proses induk kecuali dalam hal keluaran minor, seperti jalur-jalur kesalahan, yang hanya dimasukkan pada diagram anak [4]

UNIVERSITAS
MIKROSKIL



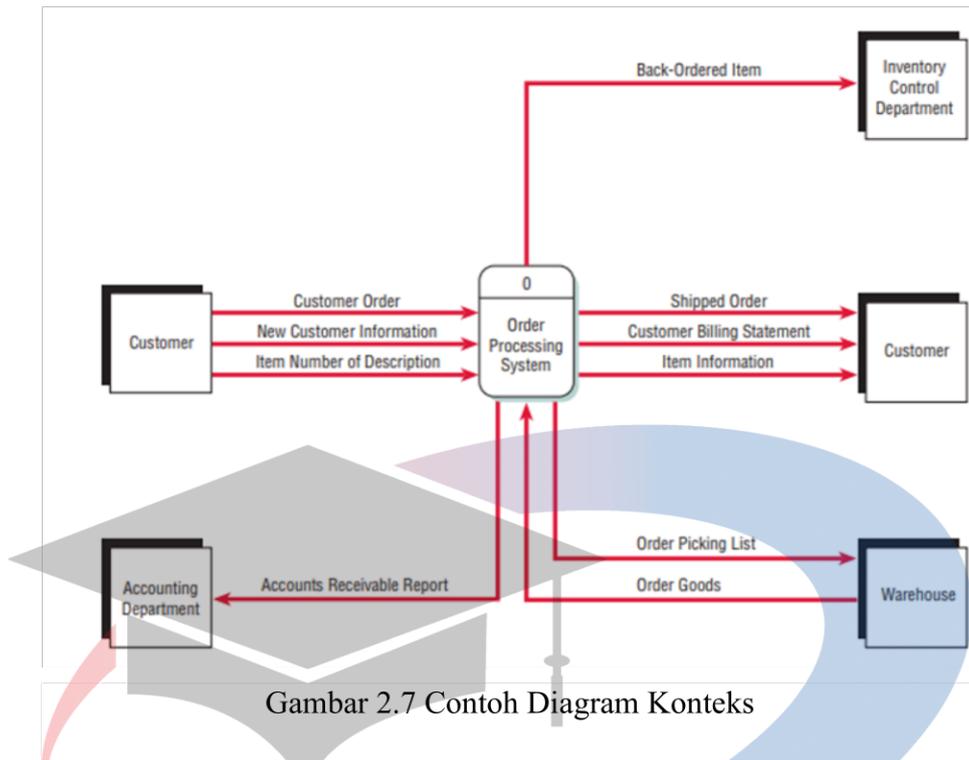
Gambar 2.6 Contoh DFD (Aliran Data)

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.

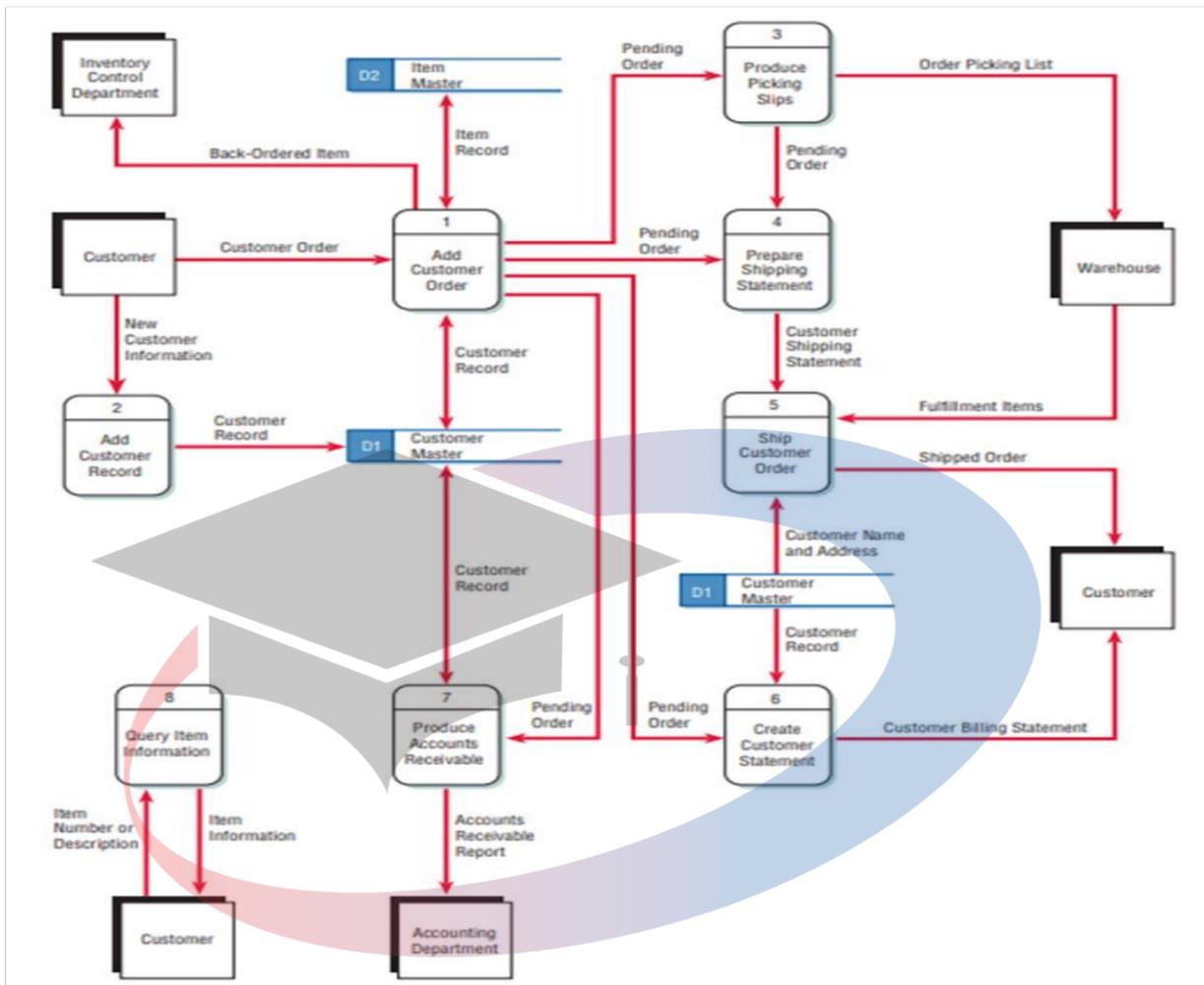
Berikut ini adalah contoh penggambaran diagram konteks [7]:



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* [7]:

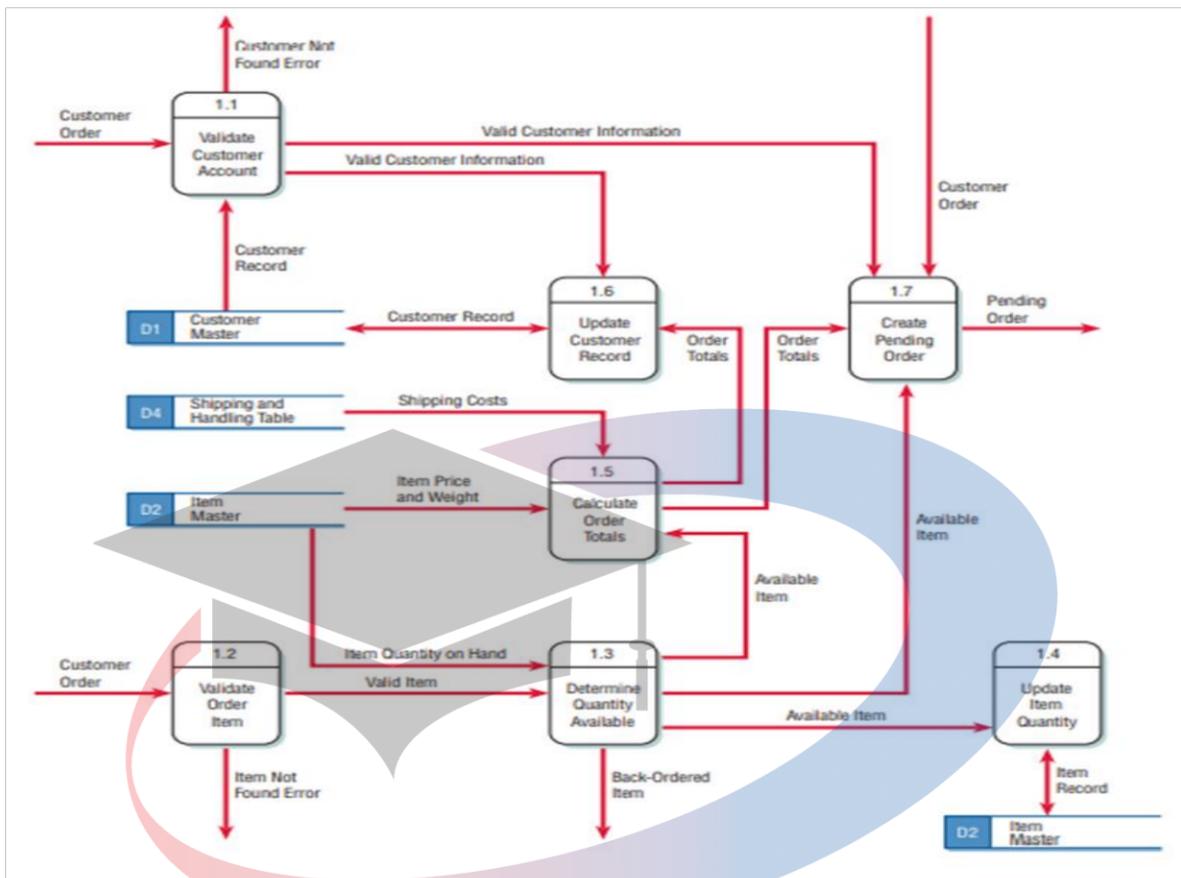


Gambar 2.8 Contoh Diagram Level Nol

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* anak [7]:



Gambar 2.9 Contoh Diagram Anak

2.3.3 Kamus Data

Kamus Data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau file di dalam sistem. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi dalam basis data, juga dapat digunakan untuk [10].

1. Validasi keakuratan dan kelengkapan DFD.
2. Merencanakan *user interface* baik *input*, dan *output*.
3. Menentukan muatan data yang disimpan dalam file-file.
4. Mengembangkan logika untuk proses-proses DFD.

Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dapat pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi

yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, Kamus data digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD. Kamus data mendefinisikan elemen-elemen data dengan fungsi sebagai berikut [10].

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.
2. Mendefinisikan komposisi paket data yang bergerak melalui aliran (misalnya alat diuraikan menjadi kota, negara dan kode pos).
3. Mendeskripsikan komposisi penyimpanan data.
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*).

Tabel 2.1 Simbol-Simbol Kamus Data

No.	Simbol	Uraian
1.	=	Terdiri dari, mendefinisikan, diuraikan menjadi
2.	+	Dan
3.	()	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field <i>numeric</i> pada struktur file.
4.	{ }	Menunjukkan elemen-elemen <i>repetitive</i> , juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau berapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.
5.	[]	Menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada seara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah atau sama lain (dengan kata lain, memiliki salah satu dari jumlah alternatif, seleksi).
6.		Pemisah sejumlah alternatif pilihan antara simbol [].
7.	@	Identifikasi atribut kunci.

8.	**	Komentar.
----	----	-----------

Berikut adalah contoh kamus data:

Tabel 2.2 Contoh Kamus Data

Pesanan Konsumen	=	Nomor Konsumen + Nama Konsumen + Alamat + Telepon + Nomor Katalog + Tanggal Pesanan + {Item Pesanan Yang Tersedia} + Total Barang + (Pajak) + Pengiriman dan Penanganan + Total Pesanan + Metode Pembayaran + (Jenis Kartu Kredit) + (Nomor Kartu Kredit) + (Masa Berlaku)
Nama Konsumen	=	Nama Pertama + (Inisial Nama Tengah) + Nama Keluarga
Alamat	=	Jalan + (Apartemen) + Kota + Negara Bagian + Kode Pos + (Panjang Kode Pos) + (Negara)
Telepon	=	Kode Area + Nomor Lokal
Item Pesanan Yang Tersedia	=	Jumlah Yang Dipesan + Nomor Item + Deskripsi Item + Ukuran + Warna + Harga + Total Item
Metode Pembayaran	=	[Cek Utang Wesel]
Jenis Kartu Kredit	=	[World's Trend American Express Master Card Visa]

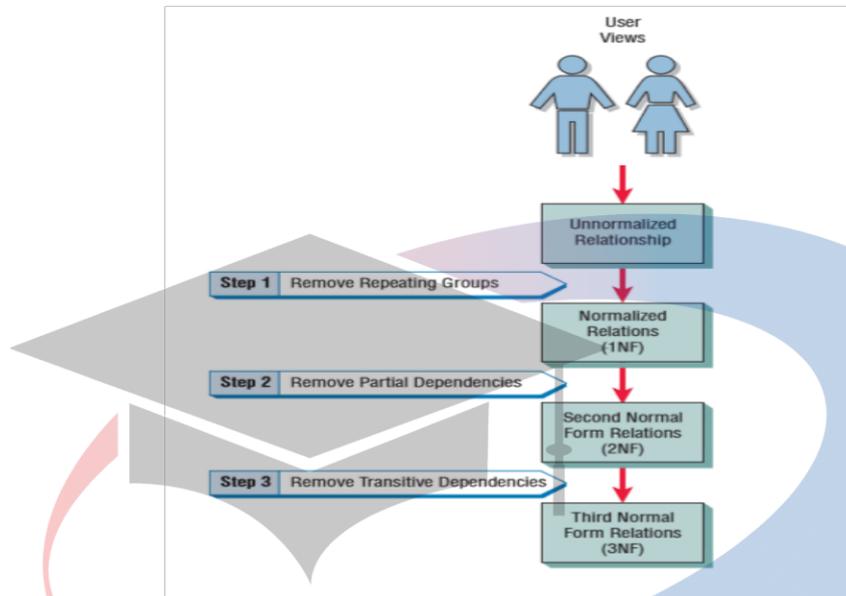
2.3.4 Normalisasi

Normalisasi adalah transformasi dari pandangan pengguna yang kompleks dan penyimpanan data ke satu set struktur data yang lebih kecil dan stabil. Selain lebih sederhana dan lebih stabil, struktur data yang dinormalisasi lebih mudah dipelihara dari pada struktur data lainnya [7].

Dimulai dengan tampilan pengguna atau penyimpanan data yang dikembangkan untuk kamus data, analisis menormalkan struktur data dalam tiga langkah. Setiap langkah melibatkan prosedur penting, yang menyederhanakan struktur data. Hubungan berasal dari tampilan pengguna atau penyimpanan data kemungkinan besar tidak akan dinormalisasi. Tahap pertama dari proses ini termasuk menghapus semua kelompok berulang dan mengidentifikasi kunci utama. Untuk melakukannya, hubungan tersebut perlu dipecah menjadi dua atau lebih [7].

Pada titik ini, hubungan tersebut mungkin sudah dari bentuk normal ketiga, tetapi kemungkinan langkah lebih lanjut akan diperlukan untuk mengubah hubungan ke bentuk

normal ketiga. Langkah kedua memastikan bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua dependensi parsial dihapus dan ditempatkan di hubungan lain. Langkah ketiga menghapus semua dependensi transitif. Ketergantungan transitif adalah atribut di mana atribut bukan kunci bergantung pada atribut bukan kunci lainnya [7]. Berikut gambar tahap normalisasi:



Gambar 2.10 Tahapan Normalisasi

Tujuan utama dari proses normalisasi adalah menyederhanakan semua kekomplekan item data. Berikut beberapa bentuk normalitas [7].

1. Bentuk Normalisasi Pertama (1NF)

Langkah pertama dalam menormalkan suatu hubungan adalah menghapus grup berulang. Dalam contoh, hubungan *SALES-REPORT* yang tidak dinormalisasi akan dipecah menjadi dua hubungan yang terpisah. Hubungan baru ini akan dinamai *SALESPERSON* dan *SALESPERSON-CUSTOMER*. Gambar 2.5 menunjukkan bagaimana hubungan asli, yang tidak dinormalisasi, *SALES-REPORT* dinormalisasi dengan memisahkan hubungan menjadi dua hubungan baru. Perhatikan bahwa hubungan *SALESPERSON* berisi kunci utama *SALESPERSON-NUMBER* dan semua atribut yang tidak diulang (*SALESPERSON-NAME* dan *SALES-AREA*).

Hubungan kedua, *SALESPERSON-CUSTOMER*, berisi kunci utama dari relasi *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*,

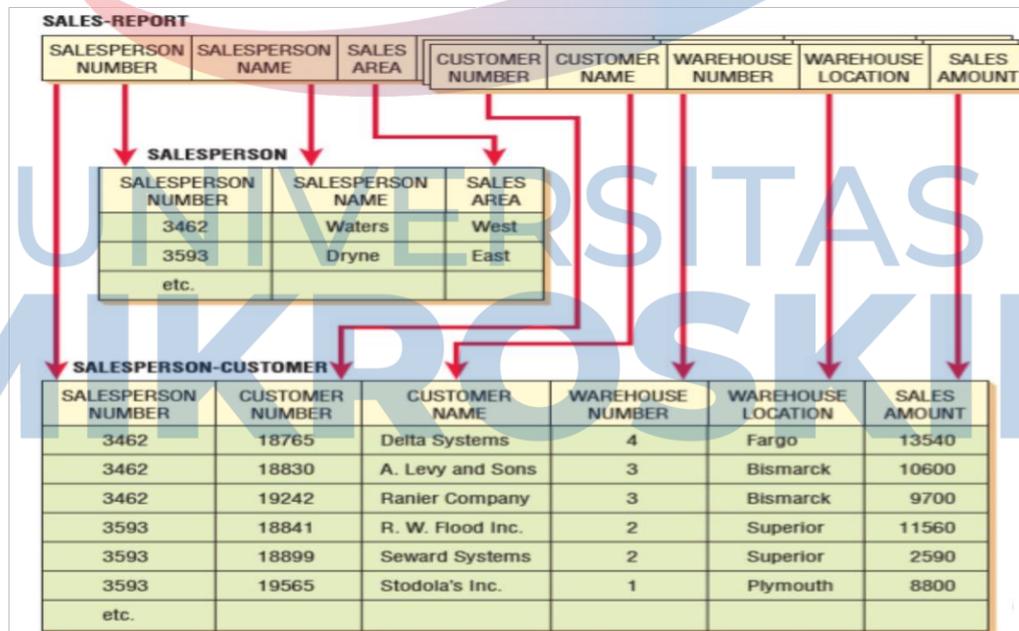
bagaimanapun, tidak secara otomatis berarti bahwa Anda akan mengetahui *CUSTOMER-NAME*, *SALES-AMOUNT*, *WAREHOUSE-LOCATION*, dan sebagainya. Dalam hubungan ini, seseorang harus menggunakan kunci gabungan (baik *SALESPERSON-NUMBER* dan *CUSTOMER-NUMBER*) untuk mengakses informasi lainnya. Hal ini dimungkinkan untuk menulis hubungan dalam notasi singkat sebagai berikut:

SALESPERSON (*SALESPERSON NUMBER*,
SALESPERSON-NAME, *SALES AREA*)

and

SALESPERSON-CUSTOMER (*SALESPERSON-NUMBER*,
CUSTOMER-NUMBER,
CUSTOMER-NAME,
WAREHOUSE-NUMBER,
WAREHOUSE-LOCATION,
SALES-AMOUNT)

Berikut gambar normalisasi 1NF:



Gambar 2.11 Normalisasi (1NF)

2. Bentuk Normalisasi Kedua (2NF)

Dalam bentuk normal kedua, semua atribut akan secara fungsional tergantung pada kunci utama. Oleh karena itu, langkah selanjutnya adalah menghapus semua atribut yang sebagian tergantung dan menempatkannya di relasi lain. Gambar 2.12 menunjukkan bagaimana hubungan *SALESPERSON-CUSTOMER* dipecah menjadi dua hubungan baru: *SALES* dan *CUSTOMER-WAREHOUSE*. Hubungan-hubungan ini juga dapat dinyatakan sebagai berikut:

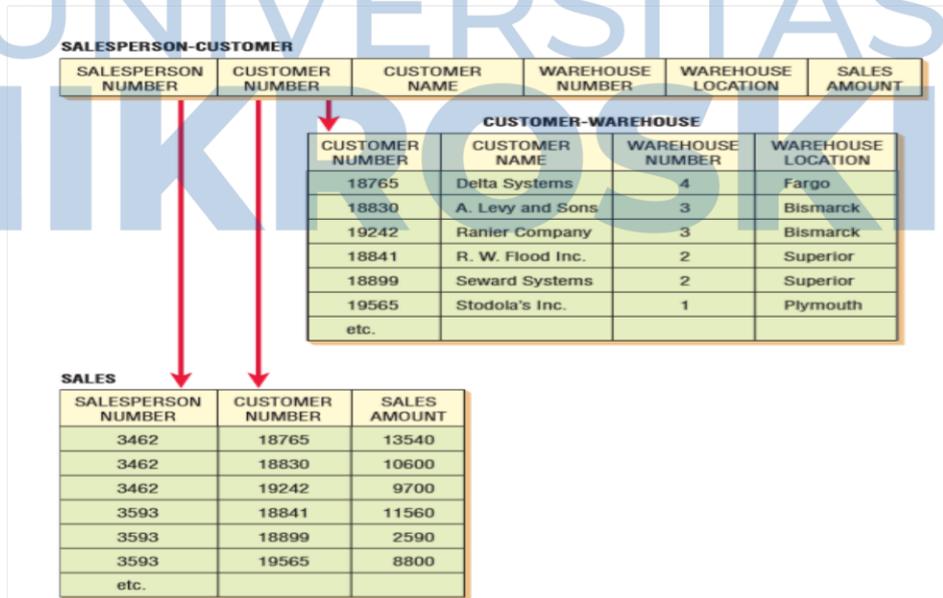
SALES (*SALESPERSON-NUMBER*, *CUSTOMER-NUMBER*,
SALES-AMOUNT)

and

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

Hubungan *CUSTOMER-WAREHOUSE* dalam bentuk normal kedua. Masih bisa disederhanakan lebih lanjut karena ada dependensi tambahan dalam relasi. Beberapa atribut bukan kunci tidak hanya bergantung pada kunci utama, tetapi juga pada atribut bukan kunci. Ketergantungan ini disebut sebagai ketergantungan transitif.

Bentuk normalisasi kedua (2NF) yang dihasilkan lebih jelas dilihat pada gambar dibawah:



Gambar 2.12 Normalisasi (2NF)

3. Bentuk Normalisasi Ketiga (3NF)

Hubungan yang dinormalisasi adalah dalam bentuk normal ketiga jika semua atribut non-kunci sepenuhnya bergantung secara fungsional pada kunci primer dan tidak ada dependensi transitif (non-kunci). Dengan cara yang mirip dengan langkah-langkah sebelumnya, adalah mungkin untuk memecah hubungan *CUSTOMER-WAREHOUSE* menjadi dua hubungan.

Dua hubungan baru disebut *CUSTOMER* dan *WAREHOUSE*, dan dapat ditulis sebagai berikut:

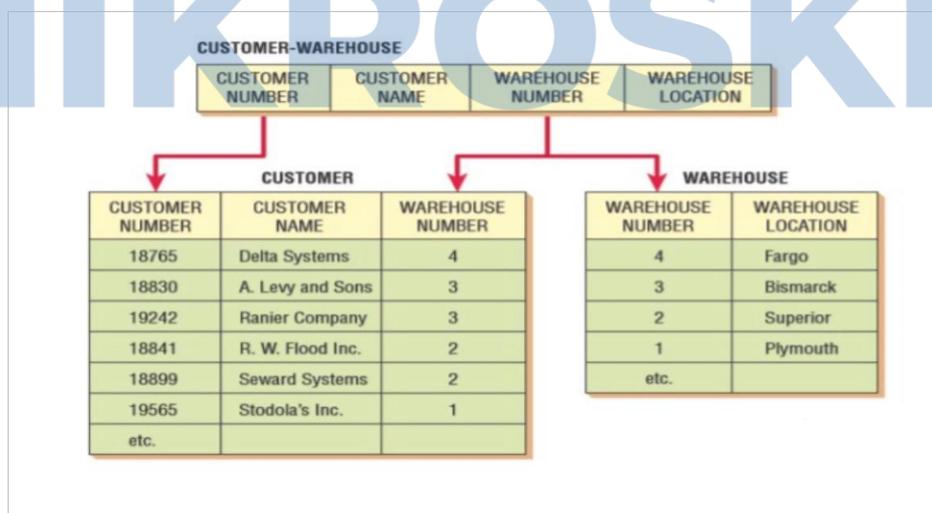
CUSTOMER (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*,
WAREHOUSE-NUMBER)

and

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

Kunci utama untuk hubungan *CUSTOMER* adalah *CUSTOMER-NUMBER*, dan kunci utama untuk hubungan *WAREHOUSE* adalah *WAREHOUSE-NUMBER*. Selain kunci-kunci utama ini, kami dapat mengidentifikasi *WAREHOUSE-NUMBER* menjadi kunci asing dalam hubungan *CUSTOMER*. Kunci asing adalah atribut yang non kunci dalam satu relasi tapi kunci utama dalam hubungan lain. Kami menunjuk *WAREHOUSE-NUMBER* sebagai kunci asing dalam notasi sebelumnya dan dalam angka dengan menggarisbawahi dengan garis putus-putus.

Berikut bentuk normalisasi ketiga (3NF) yang dihasilkan lebih jelas dapat dilihat pada gambar dibawah ini:



Gambar 2.13 Normalisasi (3NF)

Akhirnya, hubungan *SALES-REPORT* yang asli dan tidak dinormalisasi telah diubah menjadi empat hubungan 3NF. Dalam meninjau hubungan yang ditunjukkan pada Gambar 2.14, kita dapat melihat bahwa hubungan tunggal *SALES-REPORT* diubah menjadi empat hubungan berikut:

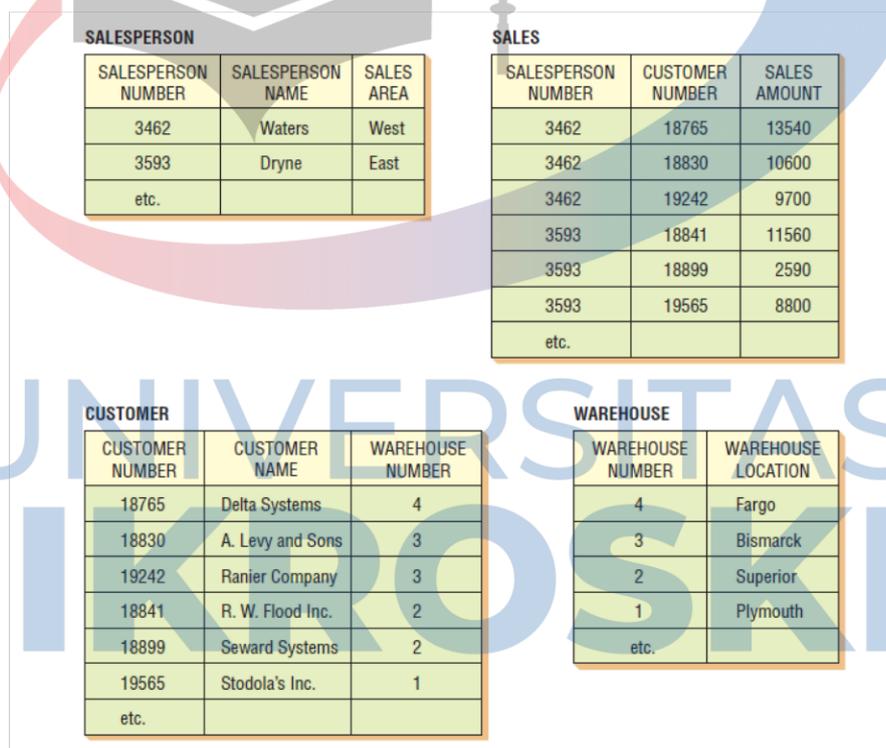
SALESPERSON (*SALESPERSON-NUMBER*, *SALESPERSON-NAME*,
SALES-AREA)

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

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

and

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



Gambar 2.14 Basis Data Lengkap pada Empat Hubungan

Bentuk normalisasi ketiga cukup untuk sebagian besar masalah desain *database*. Penyederhanaan yang diperoleh dari mengubah sebuah hubungan yang tidak dinormalisasi menjadi seperangkat hubungan 3NF adalah manfaat luar biasa ketika tiba saatnya untuk memasukkan, menghapus, dan memperbarui informasi dalam *database* [7].

2.4 Basis Data

Basis data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [11].

Sebagai satu kesatuan istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti: [11]

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik. Untuk selanjutnya di dalam buku ini, kita akan menggunakan istilah Tabel (*Table*), sebagai komponen utama pembangunan basis data.

Basis Data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik seperti cakram magnetis (*magnetic disk* atau disingkat sebagai *disk* saja). Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola oleh manusia, sementara basis data dikelola melalui perantara mesin pintar elektronik (yang kita kenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metode yang dapat digunakan dalam upaya penyimpanan [11].

2.5 Penjualan

Penjualan adalah kegiatan sejak diterimanya pesanan dari pembeli, pengiriman barang, pembuatan faktur (penagihan), dan pencatatan penjualan atau suatu kegiatan yang dilakukan manusia untuk menyampaikan barang kebutuhan yang telah dihasilkan kepada mereka yang memerlukannya dengan imbalan uang menurut harga yang ditentukan [10].

Kegiatan penjualan terdiri atas penjualan barang dan jasa, baik secara kredit maupun secara tunai. Dalam transaksi penjualan kredit, jika pesanan dari pelanggan telah dipenuhi dengan pengiriman barang atau penyerahan jasa untuk jangka waktu tertentu perusahaan memiliki piutang kepada pelangganya. Dalam sistem penjualan secara tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli [12].

Berikut adalah penjelasan mengenai dua jenis penjualan, antara lain:

1. Transaksi penjualan kredit, jika *order* dari pelanggan telah dipenuhi dengan pengiriman barang atau penyerahan jasa. Kegiatan penjualan secara kredit ini ditangani oleh perusahaan melalui sistem penjualan kredit.
2. Transaksi penjualan tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli. Kegiatan penjualan secara tunai ini ditangani oleh perusahaan melalui sistem penjualan tunai. Kegiatan penjualan yang akan diuraikan hanya terkait dengan kegiatan penjualan barang [10].

Fungsi-fungsi yang terkait dengan penjualan yaitu:

1. Fungsi penjualan
Fungsi ini bertanggung jawab untuk menerima *order* dan pembelian, mengisi faktur penjualan tunai dan menyerahkan faktur tersebut kepada pembeli untuk kepentingan pembayaran harga barang ke fungsi kas.
2. Fungsi kas
Fungsi ini bertanggung jawab sebagai penerima kas dari pembeli, mencatat jumlah penerimaan kas, serta menyiapkan laporan penerimaan kas ke fungsi akuntansi.
3. Fungsi gudang
Fungsi ini bertanggung jawab menyediakan barang yang diperlukan oleh pelanggan sesuai dengan yang tercantum dalam tembusan faktur penjualan yang diterima dari fungsi penjualan.
4. Fungsi pengiriman
Fungsi ini bertanggung jawab untuk membungkus barang dan menyerahkan barang yang telah dibayar harganya kepada pembeli.

5. Fungsi akuntansi

Fungsi ini bertanggung jawab sebagai pencatat transaksi penjualan dan sebagai pembuat laporan penjualan.

Dalam transaksi penjualan, tidak semua penjualan mendapatkan pendapatan (*revenue*) bagi perusahaan. Ada kalanya pembeli mengembalikan barang yang telah dibelinya kepada perusahaan. Transaksi pengembalian barang oleh pembeli ini ditangani perusahaan melalui sistem retur penjualan [10].

2.6 Pembelian

Sistem akuntansi pembelian digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan. Transaksi pembelian dapat digolongkan menjadi dua: pembelian lokal dan impor. Pembelian impor dalam pembelian dari pemasok dalam negeri, sedangkan impor adalah pemasok dari luar negeri. Pembahasan sistem akuntansi pembelian ini diterapkan dalam manufaktur sebagai model [10]. Pembelian sama pentingnya dengan penjualan, yaitu untuk memenuhi kebutuhan setiap perusahaan, seperti kebutuhan peralatan kantor, gedung, peralatan produksi, dan lain sebagainya [13].

Fungsi yang terkait dalam sistem akuntansi pembelian adalah:

1. Fungsi gudang

Dalam sistem akuntansi pembelian, fungsi gudang bertanggung jawab untuk mengajukan permintaan pembelian sesuai dengan posisi persediaan yang ada di gudang dan untuk menyimpan barang yang telah diterima oleh fungsi penerimaan. Untuk barang-barang yang langsung pakai (tidak ada persediaan barang di gudang), permintaan pembelian diajukan oleh pemakai barang.

2. Fungsi pembelian

Fungsi pembelian bertanggung jawab untuk memperoleh informasi mengenai harga barang, menentukan pemasok yang dipilih dalam pengadaan barang, dan mengeluarkan *order* pembelian kepada pemasok yang dipilih.

3. Fungsi penerimaan

Dalam sistem akuntansi pembelian, fungsi ini bertanggung jawab untuk melakukan pemeriksaan terhadap jenis, mutu, dan kuantitas barang yang diterima dari pemasok guna menentukan apakah barang tersebut dapat diterima atau tidak oleh perusahaan.

Fungsi ini juga bertanggung jawab untuk menerima barang dari pembeli yang berasal dari transaksi retur penjualan.

4. Fungsi akuntansi

Fungsi akuntansi yang terkait dalam transaksi pembelian adalah fungsi pencatat utang dan fungsi pencatat persediaan. Dalam sistem akuntansi pembelian, fungsi pencatat utang bertanggung jawab untuk mencatat transaksi pembelian ke dalam *register* bukti kas keluar dan untuk menyelenggarakan kartu utang sebagai buku pembantu utang [10].

Secara garis besar transaksi pembelian mencakup prosedur berikut ini:

1. Fungsi gudang mengajukan permintaan pembelian ke fungsi pembelian.
2. Fungsi pembelian meminta penawaran harga dari berbagai pemasok.
3. Fungsi pembelian menerima penawaran harga dari berbagai pemasok dan melakukan pemilihan pemasok.
4. Fungsi pembelian membuat *order* pembelian kepada pemasok yang dipilih.
5. Fungsi penerimaan memeriksa dan menerima barang yang dikirim oleh pemasok.
6. Fungsi penerimaan menyerahkan barang yang diterima kepada fungsi gudang untuk disimpan.
7. Fungsi penerimaan melaporkan penerimaan barang kepada fungsi akuntansi
8. Fungsi akuntansi menerima faktur taguhan dari pemasok dan atas dasar faktur dari pemasok tersebut, fungsi akuntansi mencatat kewajiban yang timbul dari transaksi pembelian [10].

2.7 Persediaan

Persediaan adalah salah satu asset perusahaan. Selain itu, persediaan barang merupakan salah satu aktifitas perusahaan yang sangat penting bagi perkembangan perusahaan. Permasalahan yang sering dihadapi oleh perusahaan ini adalah sering terjadi kesalahan dalam pencatatan data transaksi pemesanan dan penjualan barang, sehingga kesulitan dalam pengontrolan persediaan barang. Semakin bertambahnya jumlah jenis barang, timbul beberapa permasalahan yaitu informasi persediaan barang tidak dapat disajikan dengan cepat, tepat dan akurat. Adapun penyebab munculnya permasalahan tersebut adalah pengolahan data transaksi yang membutuhkan beberapa tahapan dan sering terjadi kesalahan pencatatan dalam faktur, *form* serta laporan yang dibuat. Selain itu pengolahan data transaksi menjadi informasi persediaan barang sering ditunda oleh staf di

bagian persediaan barang. Untuk mengatasi permasalahan tersebut, maka diperlukan sistem informasi persediaan barang yang tepat dan sesuai kebutuhan [14].

Dalam perusahaan manufaktur, terdiri dari: persediaan produk jadi, persediaan produk dalam proses, persediaan bahan baku, persediaan bahan penolong, persediaan perlengkapan pabrik, dan persediaan suku cadang. Dalam perusahaan dagang, persediaan hanya terdiri dari satu jenis, yaitu persediaan barang dagang, yang merupakan barang yang dibeli untuk dijual kembali. Transaksi yang mengubah persediaan perlengkapan pabrik, dan persediaan suku cadang, terkait dengan transaksi intern perusahaan dan transaksi yang terkait dengan pihak luar perusahaan (penjualan dan pembelian), sedangkan transaksi yang merubah persediaan produk dalam proses seluruhnya berupa transaksi intern perusahaan [10].

Secara filosofi, persediaan diperlukan untuk menghadapi dan mengantisipasi beberapa situasi. Pertama, berkenaan dengan ketidakpastian. Terdapat ketidakpastian di dalam bisnis baik dalam sisi permintaan maupun penawaran. Dari sisi permintaan, jumlah yang dikehendaki pelanggan bervariasi dan tidak diketahui secara pasti [15].

2.8 Hutang

Hutang merupakan kewajiban perusahaan kepada pihak lain yang belum terpenuhi dan merupakan sumber dana atau modal perusahaan [16]. Menurut Kieso (2011:238), Kewajiban jangka panjang yaitu terdiri dari pengorbanan ekonomi yang sangat mungkin di masa deoan akibat kewajiban sekarang yang tidak dibayarkan dalam satu tahun atau siklus operasi perusahaan. Berdasarkan definisi diatas dapat disimpulkan bahwa liabilitas tidak lancar atau kewajiban jangka Panjang adalah pinjaman yang diperoleh dari pihak ketiga yang pelunasannya atau jatuh temponya lebih dari satu periode akuntansi(1 tahun) dan besar hutang jangka panjang tersebut tidak boleh lebih dari jumlah modal sendiri [17]. Hutang usaha merupakan perbuatan pinjam-meminjam yang dilakukan oleh seseorang atau perusahaan yang dimana diwajibkan untuk melunasi hutang kepada kreditur sebagaimana telah diatur dalam perjanjian [18]. Secara umum ada 2 jenis hutang yakni utang untuk waktu yang panjang dan juga utang untuk waktu yang pendek. Utang adalah beban yang harus diselesaikan oleh perusahaan terhadap pihak lain yang disebabkan karna adanya kegiatan transaksi yang pernah terjadi. Keadaan perusahaan yang mampu menyanggupi kewajiban dan posisi aktiva yang tepat berada dalam kondisi stabil. Jika aktiva perusahaan lebih kecil dari kewajibannya maka kefailitan bisa terjadi akibat meningkatnya pembayaran beban bunga dan kewajiban dasar dalam jumlah besar [19].

Biasanya perusahaan yang memilih menggunakan hutang akan memiliki tingkat risiko yang tinggi atas beban bunga yang dibebankan oleh kreditur. Perusahaan perlu menganalisis *ratio leverage* agar perusahaan dapat menjaga batas aman penggunaan hutang. *Debt to Equity Ratio* menunjukkan pengukur tingkat penggunaan hutang (total hutang) terhadap modal yang dimiliki perusahaan. Apabila biaya yang ditimbulkan oleh pinjaman (*cost of debt*) lebih kecil daripada biaya modal sendiri (*cost of equity*), maka sumber dana yang berasal dari pinjaman atau hutang akan lebih efektif dalam menghasilkan laba [20].

Hutang juga bisa disebut sebagai salah satu sumber pembiayaan eksternal yang digunakan oleh perusahaan untuk membiayai kebutuhan dananya dengan kata lain hutang adalah semua kewajiban keuangan perusahaan kepada pihak lain yang belum terpenuhi, dimana hutang ini merupakan sumber dana atau modal perusahaan yang berasal dari kreditur [21].

Memiliki hutang jangka panjang selain menguntungkan dan dapat memberikan manfaat kepada perusahaan, namun juga memiliki beberapa risiko. Beberapa risiko hutang jangka Panjang diantaranya adalah:

1. Semakin lama jangka waktu pinjaman dana dan peluasannya maka risiko juga semakin tinggi
2. Hutang merupakan beban tetap yang harus ditanggung oleh perusahaan
3. Memiliki tanggal waktu jatuh tempo pembayaran hutang yang sudah pasti
4. Hanya dapat memperoleh sumber dana yang terbatas dari hasil pinjaman kemungkinan nilai saham perusahaan akan turun akibat tingkat tinggi atau rendah jumlah pinjaman [17].

UNIVERSITAS
MIKROSKIL