

BAB II

TINJAUAN PUSTAKA

2.1. Konsep Sistem Informasi

2.1.1 Sistem

Sistem merupakan kumpulan bagian-bagian ataupun komponen yang saling bergantung satu sama lain dan terpadu untuk mencapai satu tujuan. Model dasar dari bentuk sistem adalah terdapat masukan, pengolahan, dan keluaran, yang kemudian dapat dikembangkan hingga menyertakan media penyimpanan [1]. Sistem juga dapat diartikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Pendekatan sistem jaringan kerja dari prosedur lebih menekankan lebih menekankan urutan-urutan operasi dalam sistem [2].

Suatu sistem memiliki karakteristik atau sifat-sifat khas yang menggambarkan sistem tersebut. Adapun karakteristik sistem dijabarkan sebagai berikut [1]:

1. Komponen Sistem (*Components*)

Sistem terdiri dari sejumlah komponen yang saling berinteraksi atau bekerja sama membentuk satu kesatuan. Komponen-komponen tersebut dapat berupa suatu subsistem yang memiliki sifat yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*Boundary*)

Karakteristik ini merupakan daerah yang membatasi antara sistem dengan sistem yang lain. Batasan sistem memungkinkan sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem ini merupakan bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem. Ini dapat menguntungkan atau merugikan sistem. Walaupun begitu, harus tetap dijaga dan dipelihara agar tidak mengganggu keberlangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Sistem juga memiliki media yang menghubungkannya dengan subsistem lain yang disebut dengan penghubung sistem. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung ini sehingga membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Input adalah energi yang dimasukkan ke dalam sistem, dapat berupa pemeliharaan dan sinyal. Misalnya, dalam unit sistem komputer, “program” adalah pemeliharaan yang digunakan untuk mengoperasikan komputernya dan “data” adalah sinyal yang diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Energi yang telah diolah dan diklasifikasikan tersebut menghasilkan keluaran yang berguna bagi subsistem yang lain. Keluaran yang dihasilkan adalah informasi yang dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal lain yang menjadi masukan bagi subsistem lain.

7. Pengolah Sistem (*Process*)

Proses yang terdapat pada sistem akan mengubah masukan menjadi keluaran. Misalnya, sistem akuntansi yang akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen perusahaan.

8. Sasaran Sistem (*Objective*)

Sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Sistem yang tidak memiliki sasaran tidak akan berguna. Suatu sistem akan dikatakan berhasil jika mengenai sasaran atau tujuan yang telah direncanakan.

2.1.2 Informasi

Informasi merupakan data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sumber dari informasi adalah data. Nilai dari informasi tersebut penting dalam sebuah keputusan [1]. Informasi juga dapat diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya [2].

Informasi bernilai jika manfaat lebih efektif dibandingkan dengan biaya yang diperlukan untuk mendapatkannya. Namun, informasi pada umumnya sulit untuk

dihubungkan antara manfaat dengan biayanya. Sebagian informasi juga tidak dapat ditafsirkan dengan pasti keuntungannya dengan suatu nilai uang, tetapi dapat ditafsir dengan nilai efektivitas. Pengukuran nilai informasi pada umumnya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*. Nilai informasi dapat didasarkan atas sepuluh sifat, yaitu [1]:

1. Mudah diperoleh, menunjukkan informasi dapat diperoleh dengan mudah dan cepat.
2. Luas dan lengkap, menunjukkan kelengkapan isi informasi. Sifat ini tidak hanya terkait volumenya, tetapi juga terkait keluaran informasi.
3. Ketelitian, menunjukkan minimnya kesalahan dalam informasi. Dalam keterkaitannya dengan volume data yang besar biasanya dapat terjadi dua jenis kesalahan, yaitu kesalahan pencatatan dan kesalahan perhitungan.
4. Kecocokan, menunjukkan seberapa baik keluaran informasi dalam hubungan dengan permintaan para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi.
5. Ketepatan waktu, menunjukkan tidak adanya keterlambatan jika informasi perlu didapatkan.
6. Kejelasan, menunjukkan keluaran informasi yang bebas dari ketidakjelasan.
7. Keluwesan, berhubungan dengan apakah keluaran informasi dapat disesuaikan tidak hanya dengan beberapa keputusan, tetapi juga dengan beberapa pengambil keputusan.
8. Dapat dibuktikan, menunjukkan kemampuan beberapa pemakai informasi untuk menguji keluaran informasi dan sampai pada kesimpulan yang sama.
9. Tidak ada prasangka, berhubungan dengan tidak adanya keinginan untuk mengubah informasi agar mendapatkan kesimpulan yang telah dipertimbangkan sebelumnya.
10. Dapat diukur, menunjukkan hakikat informasi yang dihasilkan dari sistem informasi formal.

Selain dari sifat informasi, penting juga mengetahui kualitas suatu informasi. Kualitas informasi tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*). Berikut ini penjelasan dari ketiga hal tersebut:

1. Akurat (*accurate*)

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

2. Tepat waktu (*timeline*)

Hal ini mengacu kepada informasi yang tidak boleh terlambat diterima oleh penerimanya. Informasi yang telah usang tidak akan mempunyai nilai karena informasi merupakan landasan dalam pengambilan keputusan. Pengambilan keputusan yang terlambat akan berakibat fatal bagi organisasi.

3. Relevan (*relevance*)

Informasi harus mempunyai manfaat untuk pemakaiannya. Relevansi informasi dapat berbeda pada satu orang dengan yang lain.

2.1.3 Sistem Informasi

Dari pengertian sistem dan informasi sebelumnya, maka terbentuk suatu konsep sistem informasi yang merupakan sistem dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan [2].

Sistem informasi terdiri dari komponen-komponen yang disebut sebagai blok bangunan (*building block*). Blok bangunan tersebut terdiri dari enam blok. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan lainnya membentuk suatu kesatuan untuk mencapai sasaran. Adapun blok-blok tersebut, yaitu [1]:

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. *Input* ini merupakan metode dan media untuk menangkap data yang akan dimasukkan, berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok model terdiri dari kombinasi prosedur, logika, dan model matematik yang memanipulasi data *input* dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan *output* yang diinginkan.

3. Blok keluaran (*output block*)

Keluaran merupakan hasil produk dari sistem informasi berupa informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pengguna sistem.

4. Blok teknologi (*technology block*)

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

5. Blok basis data (*database block*)

Basis data merupakan kumpulan data yang saling terkait dan berhubungan, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Basis data digunakan untuk menyimpan data agar dapat menyediakan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan agar menghasilkan informasi yang berkualitas serta dapat mengefisienkan kapasitas penyimpanannya. Perangkat lunak DBMS (*Database Management System*) ini yang digunakan untuk mengakses dan memanipulasi basis data.

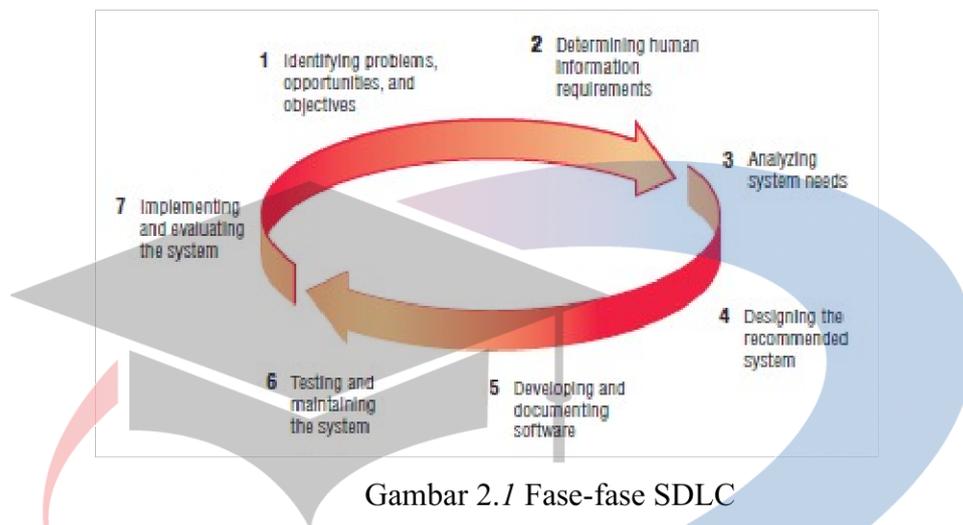
6. Blok kendali (*control block*)

Suatu sistem informasi dapat dirusak oleh beberapa hal, seperti bencana alam, api, temperatur, air, debu, kecurangan, kegagalan, sabotase, dan sebagainya. Untuk mencegah hal-hal tersebut terjadi, maka perlu untuk dilakukan perancangan dan penerapan pengendalian.

2.2. SDLC (*System Development Life Cycle*)

SDLC adalah pendekatan bertahap untuk analisis dan desain yang berpendapat bahwa sistem paling baik dikembangkan melalui penggunaan siklus analis dan kegiatan tertentu pengguna. Siklus dalam SDLC dibagi menjadi 7 fase,

seperti yang ditunjukkan pada Gambar 2.1. Meskipun setiap fase disajikan secara terpisah, namun fase-fase tersebut tidak pernah dicapai sebagai langkah yang terpisah. Sebaliknya, beberapa aktivitas dapat terjadi secara bersamaan, dan aktivitas dapat diulang [3].



1. Fase Pertama (Mengidentifikasi Masalah, Peluang, dan Tujuan)

Fase pertama mengharuskan analis melihat secara jujur apa yang terjadi dalam bisnis. Kemudian, bersama dengan anggota organisasi lainnya, analis menunjukkan masalah. Seringkali orang lain akan memunculkan masalah ini, dan itulah alasan analis awalnya dipanggil. Peluang adalah situasi yang menurut analis dapat ditingkatkan melalui penggunaan komputerisasi sistem informasi. Merebut peluang dapat memungkinkan bisnis untuk mendapatkan keunggulan kompetitif atau menetapkan standar industri.

Mengidentifikasi tujuan juga merupakan komponen penting dari fase pertama. Pertama-tama, analis harus temukan apa yang dilakukan bisnis. Kemudian analis akan dapat melihat apakah beberapa aspek aplikasi sistem informasi dapat membantu bisnis mencapai tujuannya dengan mengatasi masalah atau peluang tertentu.

Stakeholder yang terlibat dalam fase pertama adalah pengguna, analis, dan manajer sistem yang berkoordinasi dengan proyek. Kegiatan dalam fase ini terdiri dari mewawancarai 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 merangkum tujuan. Manajemen kemudian harus membuat keputusan apakah akan melanjutkan dengan yang diusulkan proyek. Tahap ini sangat penting untuk keberhasilan proyek kedepannya karena tidak ada yang mau membuang waktu untuk mengatasi masalah yang salah.

2. Fase Kedua (Menentukan Kebutuhan Informasi Perusahaan)

Fase berikutnya adalah menentukan kebutuhan informasi perusahaan dari pengguna yang terlibat, menggunakan berbagai alat untuk memahami bagaimana pengguna berinteraksi dalam konteks kerja mereka saat ini dengan sistem informasi. Analis akan menggunakan metode interaktif seperti wawancara, pengambilan sampel dan menyelidiki data keras, serta menggunakan kuesioner, bersama dengan metode yang tidak mengganggu, seperti mengamati perilaku pengambil keputusan dan lingkungan kantor mereka, dan metode yang mencakup semua, seperti pembuatan prototipe.

Dalam fase ini, analis berusaha untuk memahami informasi apa yang dibutuhkan pengguna untuk melakukan pekerjaan mereka. Pada titik ini, analis sedang memeriksa bagaimana cara untuk membuat sistem berguna bagi orang-orang yang terlibat. Bagaimana sistem dapat mendukung individu dengan lebih baik, serta tugas yang perlu dilakukan? Tugas baru apa yang diaktifkan oleh sistem baru yang tidak dapat dilakukan pengguna sendiri? Bagaimana sistem baru dapat dibuat untuk memperluas kemampuan pengguna di luar apa yang disediakan sistem lama? Bagaimana analis dapat membuat sistem yang bermanfaat untuk digunakan pekerja?

Orang-orang yang terlibat dalam fase ini adalah analis dan pengguna, serta manajer operasional dan pekerja operasional. Analis sistem perlu mengetahui detail fungsi sistem saat ini, seperti siapa (orang-orang yang terlibat), apa (kegiatan bisnis), di mana (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 menjalankan bisnis menggunakan metode ini, namun harus dipertimbangkan ketika merancang sistem baru.

3. Fase Ketiga (Menganalisis Kebutuhan Sistem)

Fase berikutnya yang dilakukan analisis sistem melibatkan analisis kebutuhan sistem. Alat dan teknik khusus membantu analisis membuat penentuan kebutuhan. Alat seperti diagram aliran data (DFD) untuk memetakan *input*, proses, dan *output* dari bisnis fungsi, atau *activity diagram* atau *sequence diagram* untuk menunjukkan urutan kejadian, ilustrasi sistem dalam bentuk grafik yang terstruktur. Dari aliran data, urutan, atau diagram lainnya, kamus data dapat dikembangkan yang mencantumkan semua *item* data yang digunakan dalam sistem, serta spesifikasi mereka.

Selama fase ini analisis sistem juga menganalisis keputusan terstruktur yang dibuat. Keputusan terstruktur adalah mereka yang kondisi, kondisi alternatif, tindakan, dan tindakan aturan dapat ditentukan. Ada tiga alat utama untuk menganalisis keputusan terstruktur, yaitu bahasa Inggris terstruktur (*structured English*), tabel keputusan (*decision tables*), dan pohon keputusan (*decision trees*).

Pada titik ini di SDLC, analisis sistem menyiapkan proposal sistem yang merangkum apa yang telah ditemukan tentang pengguna, kegunaan, dan kegunaan sistem saat ini, menyediakan biaya-analisis manfaat dari alternatif, dan membuat rekomendasi tentang apa (jika ada) yang harus dilakukan.

Jika salah satu rekomendasi dapat diterima oleh manajemen, analisis melanjutkan proposal itu. Setiap masalah sistem adalah unik, dan tidak pernah hanya ada satu solusi yang benar. Cara di mana rekomendasi atau solusi dirumuskan tergantung pada kualitas individu dan pelatihan profesional setiap analisis dan interaksi analisis dengan pengguna dalam konteks lingkungan kerja mereka.

4. Fase Keempat (Merancang Sistem yang Direkomendasi)

Pada fase desain SDLC, analisis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk menyelesaikan desain logis dari sistem informasi. Analisis merancang prosedur pengguna untuk membantu mereka memasukkan data secara akurat sehingga data yang masuk ke sistem informasi benar. Selain itu, analisis menyediakan bagi pengguna untuk melengkapi input informasi yang efektif sistem dengan menggunakan teknik dengan *form* yang baik dan halaman web atau layer desain.

Antarmuka pengguna dirancang dengan bantuan pengguna untuk memastikan bahwa sistem dapat didengar, terbaca, dan aman, serta menarik dan menyenangkan untuk digunakan. Fase desain juga mencakup perancangan *database* yang akan menyimpan banyak data yang dibutuhkan oleh para pengambil keputusan dalam organisasi. Pengguna mendapat manfaat dari *database* yang terorganisir dengan baik dan logis kepada mereka dan sesuai dengan cara mereka memandang pekerjaan mereka. Pada fase ini analis juga bekerja dengan pengguna untuk merancang keluaran (baik di layar atau dicetak) yang memenuhi kebutuhan informasi mereka. Terakhir, analis harus merancang kontrol dan prosedur cadangan untuk melindungi sistem dan data, serta untuk menghasilkan paket spesifikasi program untuk *programmer*.

5. Fase Kelima (Mengembangkan dan Mendokumentasikan Perangkat Lunak)

Pada fase kelima SDLC, analis bekerja dengan *programmer* untuk mengembangkan perangkat lunak yang dibutuhkan. Selama fase ini analis bekerja dengan pengguna untuk mengembangkan dokumentasi untuk perangkat lunak, termasuk manual prosedur, bantuan online, dan situs web yang menampilkan pertanyaan yang sering diajukan (*FAQ*) atau file *Read Me* yang dikirimkan dengan perangkat lunak baru. Dikarenakan pengguna terlibat sejak awal, maka fase dokumentasi harus menjawab pertanyaan yang mereka miliki dan diselesaikan bersama dengan analis. Dokumentasi memberitahu pengguna cara menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi masalah perangkat lunak.

Programmer memiliki peran kunci dalam fase ini karena mereka merancang, membuat kode, dan menghapus sintaksis kesalahan dari program komputer. Untuk memastikan kualitas, seorang *programmer* dapat melakukan salah satu desain atau panduan kode, menjelaskan bagian program yang kompleks kepada tim *programmer* yang lain.

6. Fase Keenam (Menguji dan Memelihara Sistem)

Sebelum suatu sistem informasi dapat digunakan, harus diuji terlebih dahulu. Jauh lebih hemat biaya untuk mengetahui dan menyelesaikan masalah sebelum sistem masuk ke pengguna daripada setelahnya. Beberapa pengujian diselesaikan oleh *programmer* sendiri, beberapa di antaranya oleh analis sistem

bersama dengan *programmer*. Sebuah seri tes untuk menentukan masalah dijalankan pertama dengan data sampel dan akhirnya dengan data aktual dari sistem saat ini. Seringkali rencana pengujian dibuat di awal SDLC dan disempurnakan sebagai proyek berkembang.

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 pemasok di web. Banyak prosedur sistematis yang digunakan analisis di seluruh SDLC dapat membantu memastikan bahwa pemeliharaan dijaga seminimal mungkin.

7. Fase Ketujuh (Mengimplementasi dan Mengevaluasi Sistem)

Pada fase terakhir pengembangan sistem ini, analisis membantu mengimplementasikan sistem informasi. Fase ini melibatkan pelatihan pengguna untuk menangani sistem. Vendor melakukan beberapa pelatihan, tetapi pengawasan pelatihan adalah tanggung jawab analisis sistem. Selain itu, analisis perlu merencanakan *migration* dari sistem lama ke yang baru. Proses ini termasuk mengonversi file dari format lama ke yang baru atau membangun *database*, menginstal peralatan, dan membawa yang sistem baru ke dalam lingkungan produksi (*production*).

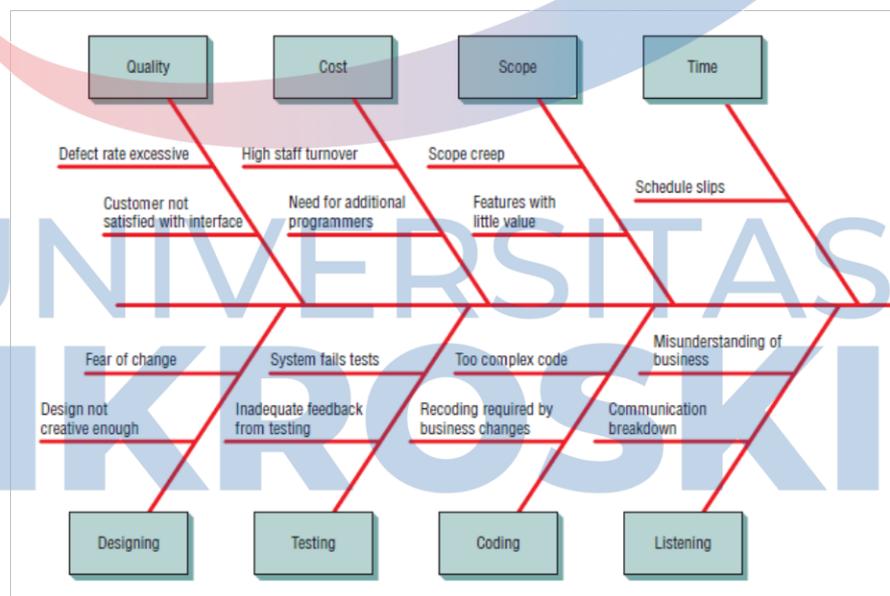
Evaluasi dimasukkan sebagai bagian dari fase akhir SDLC ini sebagian besar untuk kepentingan diskusi. Sebenarnya, evaluasi berlangsung selama setiap fase. Kriteria utama yang harus dipenuhi adalah apakah pengguna yang dituju memang menggunakan sistem. Perlu dicatat bahwa kerja sistem seringkali bersifat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

2.3. Teknik Pengembangan Sistem

2.3.1 Fishbone

Diagram *fishbone* merupakan diagram yang biasanya digambarkan oleh analis sistem untuk mengilustrasikan apa yang menjadi masalah dalam sebuah proyek. Diagram ini juga disebut sebagai diagram sebab-akibat atau diagram *Ishikawa*. Dinamakan sebagai *fishbone* karena bentuknya menyerupai tulang ikan (seperti yang ditampilkan pada Gambar 2.2).

Ketika menggambarkan diagram *fishbone* semua masalah yang mungkin terjadi akan dibuat daftar secara sistematis. Dalam pendekatan tangkas, akan berguna untuk mengatur diagram *fishbone* dengan membuat daftar semua variabel kendali sumber daya pada bagian atas dan semua aktivitas pada bagian bawah. Beberapa masalah, seperti jadwal yang terlewatkan, mungkin akan terlihat jelas, tetapi untuk masalah seperti yang adanya keinginan menambah fitur setelah analis mendengar cerita baru atau mengembangkan fitur dengan nilai kecil, akan tidak begitu jelas [3].



Gambar 2.2 Diagram fishbone

2.3.2 SWOT

Analisis SWOT adalah penilaian menyeluruh terhadap kekuatan (*strengths*), kelemahan (*weaknesses*), peluang (*opportunities*), dan ancaman (*threats*) suatu perusahaan [4]. Atau dengan kata lain, S-W-O-T digunakan untuk menilai kekuatan-keku

tan dan kelemahan-kelemahan dari sumber daya yang dimiliki perusahaan dan kesempatan eksternal dan tantangan yang dihadapi [5].

Semua organisasi memiliki kekuatan dan kelemahan dalam area fungsional bisnis, yang dapat digunakan sebagai dasar untuk tujuan dan penetapan strategi suatu organisasi [6]. Jadi, analisis SWOT merupakan instrumen yang bermanfaat dalam melakukan analisis strategi.

2.3.3 DFD (Data Flow Diagram)

Ketika analis sistem mencoba untuk memahami kebutuhan informasi pengguna, mereka harus dapat mengkonseptualisasikan bagaimana data bergerak melalui organisasi, proses atau transformasi yang dialami data, dan apa keluarannya. Meskipun wawancara dan investigasi dari data keras memberikan narasi verbal dari sistem, penggambaran visual dapat merealisasikan informasi ini bagi pengguna dan analis dengan cara yang bermanfaat [3].

Melalui teknik analisis terstruktur yang disebut *Data Flow Diagram* (DFD), seorang analis sistem dapat mengumpulkan representasi grafis dari proses data di seluruh organisasi. Dengan menggunakan kombinasi hanya empat simbol, seorang analis sistem dapat membuat gambaran dari proses yang pada akhirnya akan memberikan dokumentasi sistem yang *solid* [3].

Pendekatan aliran data, memiliki empat keunggulan utama dibandingkan penjelasan naratif tentang cara data bergerak melalui sistem [3]:

1. Bebas dari komitmen terhadap implementasi teknis sistem yang terlalu dini.
2. Pemahaman lebih lanjut tentang keterkaitan sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem saat ini kepada pengguna melalui diagram aliran data.
4. Analisis sistem yang diusulkan untuk menentukan apakah data dan proses yang diperlukan telah didefinisikan.

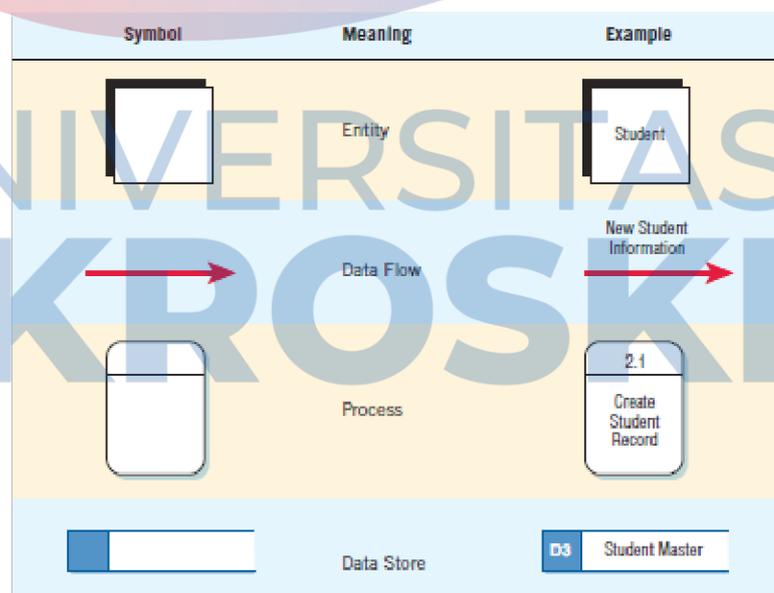
Empat simbol dasar digunakan untuk memetakan pergerakan data pada diagram aliran data, yaitu kotak ganda, dan panah, persegi panjang dengan sudut membulat, dan persegi panjang terbuka (tertutup di sisi kiri dan terbuka berakhir di sebelah kanan), seperti yang ditunjukkan pada Gambar 2.3. Seluruh sistem dan

banyak subsistem dapat digambarkan secara grafis dengan keempat simbol ini dalam kombinasi [3].

1. Kotak ganda digunakan untuk menggambarkan entitas eksternal (departemen lain, bisnis, orang, atau mesin) yang dapat mengirim data ke atau menerima data dari sistem. Entitas eksternal, atau hanya entitas, juga disebut sumber atau tujuan data, dan dianggap eksternal untuk sistem yang sedang dijelaskan. Setiap entitas diberi label dengan nama yang sesuai. Meskipun berinteraksi dengan sistem, entitas dianggap berada di luar batas sistem. Sebuah entitas harus diberi nama dengan kata benda. Entitas yang sama dapat digunakan lebih dari sekali pada aliran data tertentu diagram untuk menghindari persimpangan garis aliran data.
2. Panah menunjukkan pergerakan data dari satu titik ke titik lain, dengan kepala panah menunjuk ke arah tujuan data. Aliran data yang terjadi secara bersamaan dapat digambarkan melakukan hal itu melalui penggunaan panah paralel. Karena panah mewakili data tentang seseorang, tempat, atau benda, maka panah juga harus dijelaskan dengan kata benda.
3. Persegi panjang dengan sudut membulat digunakan untuk menunjukkan terjadinya proses transformasi. Proses selalu menunjukkan perubahan atau transformasi data. Oleh karena itu, aliran data yang meninggalkan proses selalu diberi label berbeda dari yang memasukinya. Proses mewakili pekerja dilakukan dalam sistem dan harus diberi nama menggunakan salah satu format berikut. Nama yang jelas membuatnya lebih mudah untuk memahami apa yang sedang dicapai oleh proses:
 1. Saat menamai proses tingkat tinggi, tetapkan proses tersebut dengan nama keseluruhan sistem. Contohnya adalah Sistem Kontrol Persediaan.
 2. Saat memberi nama subsistem utama, gunakan nama seperti Subsistem Pelaporan Persediaan atau Sistem Pemenuhan Pelanggan Internet.
 3. Saat menamai proses terperinci, gunakan kombinasi kata kerja, kata sifat, dan kata benda. Kata kerja menjelaskan jenis aktivitas, seperti menghitung, memverifikasi, mempersiapkan, mencetak, atau menambahkan. Kata benda menunjukkan apa hasil utama dari proses tersebut, seperti laporan atau catatan. Kata sifat menggambarkan keluaran spesifik, seperti dipesan

kembali atau persediaan, diproduksi. Contoh nama proses yang lengkap adalah “Menghitung Pajak Penjualan”.

4. Simbol dasar terakhir yang digunakan dalam diagram aliran data adalah persegi panjang terbuka, yang mewakili sebuah penyimpanan data (*data store*). Persegi panjang digambar dengan dua garis sejajar yang ditutup oleh sebuah garis pendek di sisi kiri dan terbuka berakhir di sebelah kanan. Simbol-simbol ini digambar hanya cukup lebar untuk memungkinkan mengidentifikasi huruf antara garis paralel. Dalam diagram aliran data logis, jenis penyimpanan fisik tidak ditentukan. Pada titik ini simbol *data store* hanya menunjukkan penyimpanan untuk data yang memungkinkan pemeriksaan, penambahan, dan pengambilan data. Penyimpanan data dapat mewakili penyimpanan manual, seperti lemari arsip, atau file yang terkomputerisasi atau basis data. Karena penyimpanan data mewakili orang, tempat, atau benda, mereka diberi nama dengan kata benda. Penyimpanan data sementara, seperti kertas gores atau file komputer sementara, tidak termasuk dalam diagram aliran data. Berikan setiap *data store* nomor referensi unik, seperti D1, D2, D3, dan seterusnya.



Gambar 2.3 Simbol-simbol pada DFD

Dengan pendekatan *top-down* untuk membuat diagram pergerakan data, DFD bergerak dari umum menjadi khusus. Diagram pertama membantu seorang analis sistem memahami pergerakan data dasar, tetapi sifat umumnya membatasi kegunaannya. Diagram konteks awal harus berupa gambaran umum, termasuk input

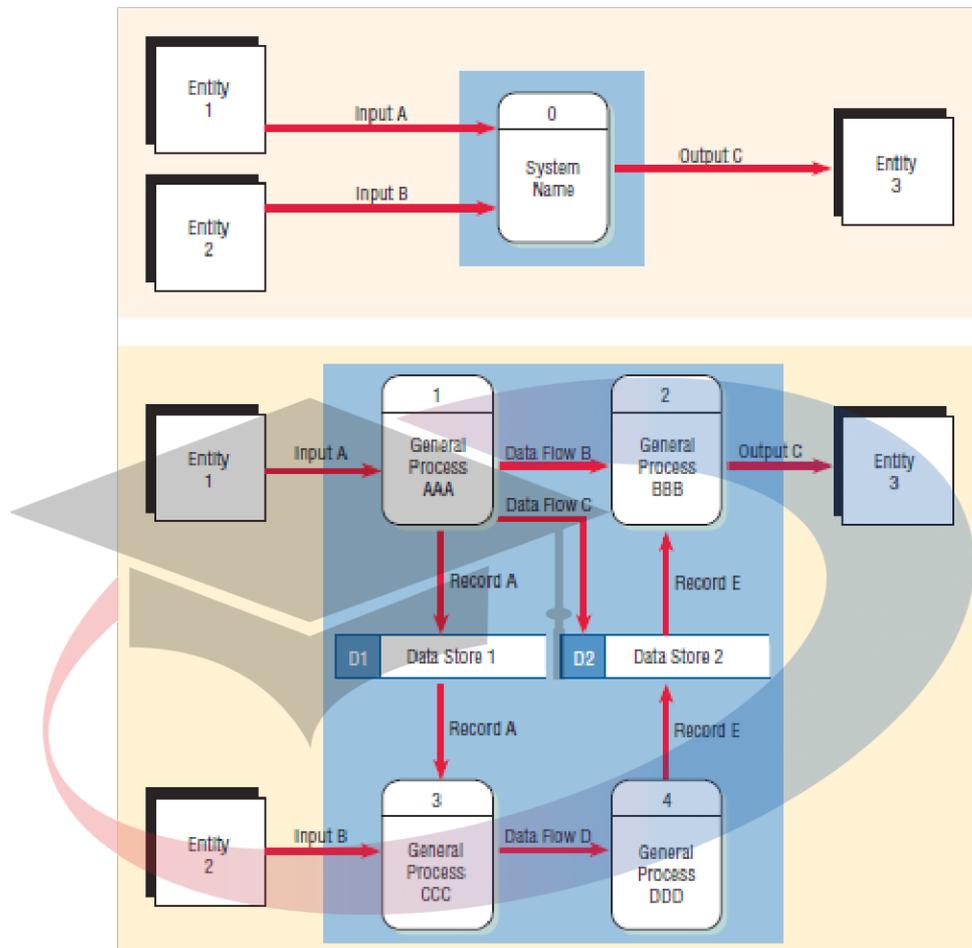
dasar, sistem umum, dan keluaran. Diagram ini akan menjadi yang paling umum, memberikan pandangan yang luar biasa pandangan pergerakan data dalam sistem dan konseptualisasi sistem seluas mungkin [3].

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. Diagramnya tidak mengandung penyimpanan data (*data store*) apa pun dan cukup sederhana untuk dibuat, setelah entitas eksternal dan aliran data ke dan dari mereka diketahui oleh analis [3].

1. Diagram 0

Lebih detail daripada yang diizinkan diagram konteks dapat dicapai dengan merincikan diagram". *Input* dan *output* yang ditentukan dalam diagram pertama tetap konstan di semua diagram berikutnya. Sisanya diagram asli, bagaimanapun, jika diteliti akan melibatkan tiga hingga sembilan proses dan menunjukkan *data store* dan aliran data tingkat rendah yang baru. Efeknya adalah mengambil kaca pembesar untuk melihat diagram aliran data asli. Setiap diagram yang diledakkan harus menggunakan hanya satu lembar kertas. Dengan membagi DFD ke dalam sub-proses, analis sistem mulai mengisi detail tentang pergerakan data. Penanganan pengecualian diabaikan untuk dua atau tiga tingkat pertama dari diagram aliran data [3].

Diagram 0 adalah perincian dari diagram konteks dan dapat mencakup hingga sembilan proses. Memasukkan lebih banyak proses pada level ini akan menghasilkan diagram yang berantakan dan sulit untuk dipahami. Setiap proses diberi nomor dengan bilangan bulat, umumnya dimulai dari sudut kiri atas diagram dan bekerja menuju sudut kanan bawah. *Data store* utama dari sistem (mewakili file *master*) dan semua entitas eksternal disertakan pada Diagram 0. Gambar 2.4 secara skematis menggambarkan diagram konteks dan Diagram 0 [3].



Gambar 2.4 Pembagian diagram konteks menjadi Diagram 0

2. Diagram Anak (Tingkatan yang Lebih Rinci)

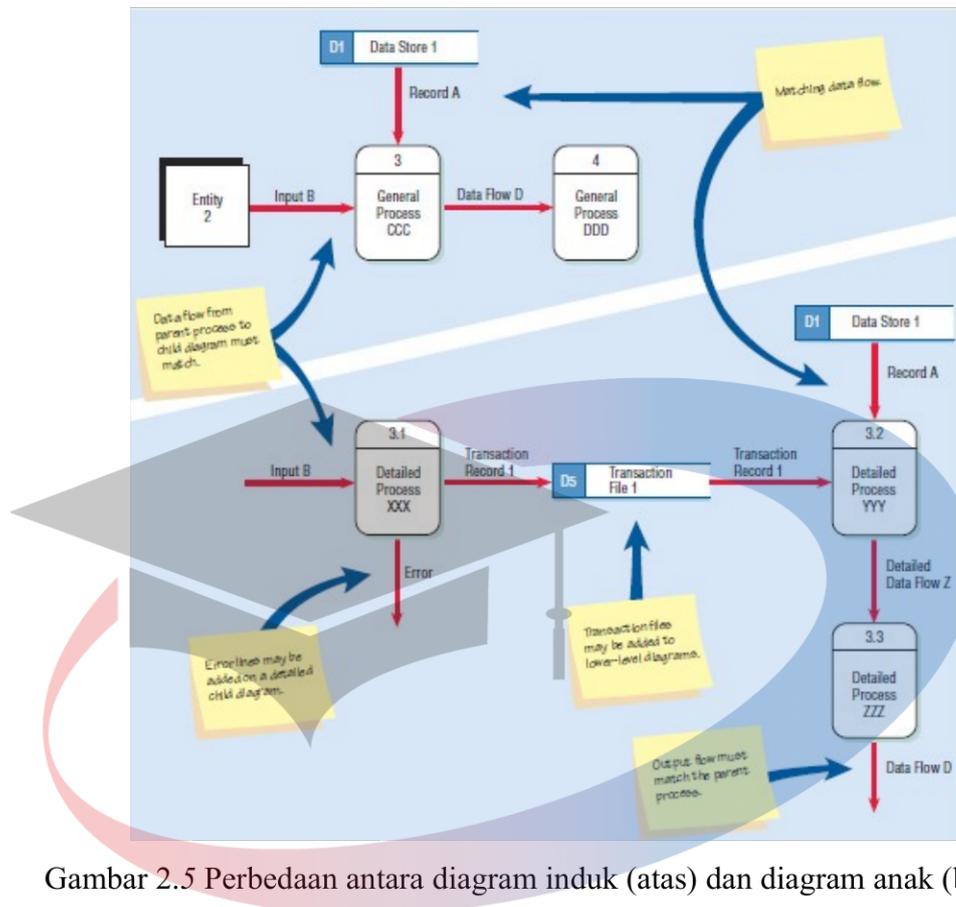
Setiap proses pada Diagram 0 pada gilirannya dapat dibagi untuk membuat diagram anak yang lebih rinci. Proses pada Diagram 0 yang dirincikan disebut proses induk (*parent process*), dan diagram yang menghasilkan disebut diagram anak (*child diagram*). Aturan utama untuk membuat diagram anak, keseimbangan vertikal, menyatakan bahwa diagram anak tidak dapat menghasilkan *output* atau menerima *input* yang dilakukan oleh proses induk tidak juga menghasilkan atau menerima. Semua aliran data masuk atau keluar dari proses induk harus ditunjukkan mengalir masuk atau keluar dari diagram anak [3].

Diagram anak diberi nomor yang sama dengan proses induknya pada Diagram 0. Misalnya, proses 3 akan dibagi ke Diagram 3. Proses pada diagram anak diberi nomor menggunakan nomor proses induk, titik desimal, dan nomor unik untuk

setiap proses anak. Pada Diagram 3, proses akan diberi nomor 3.1, 3.2, 3.3, dan seterusnya. Konvensi ini memungkinkan analisis untuk melacak serangkaian proses melalui berbagai tingkat ledakan. Jika Diagram 0 menggambarkan proses 1, 2, dan 3, diagram anak 1, 2, dan 3 semuanya berada pada level yang sama [3]. Entitas biasanya tidak ditampilkan pada diagram anak di bawah Diagram 0. Aliran data yang cocok dengan aliran induk disebut aliran data antarmuka (*interface data flow*) dan ditampilkan sebagai panah dari atau ke area kosong dari diagram anak. Jika proses induk memiliki aliran data yang terhubung ke penyimpanan data, diagram anak dapat mencakup *data store* juga. Selain itu, diagram tingkat bawah ini mungkin berisi penyimpanan data tidak ditampilkan pada proses induk. Misalnya, file yang berisi tabel informasi, seperti tabel pajak, atau file yang menghubungkan dua proses pada diagram anak dapat disertakan. Aliran data kecil, seperti garis kesalahan, dapat dimasukkan pada diagram anak tetapi tidak pada induknya [3].

Proses dapat atau tidak dapat dibagi, tergantung pada tingkat kerumitannya. Ketika sebuah proses tidak dibagi, dikatakan primitif secara fungsional dan disebut proses primitif (*primitive process*). Gambar 2.5 mengilustrasikan tingkat rinci dalam diagram aliran data anak [3].

UNIVERSITAS
MIKROSKIL



Gambar 2.5 Perbedaan antara diagram induk (atas) dan diagram anak (bawah).

2.3.4 Kamus Data (Data Dictionary)

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, *metadata*). Analisis sistem menyusun kamus data untuk memandu mereka melalui analisis dan desain. Sebuah kamus data adalah dokumen yang mengumpulkan dan menyelaraskan istilah data tertentu, dan itu menegaskan apa yang setiap istilah berarti bagi orang yang berbeda dalam organisasi. Diagram aliran data yang sebelumnya telah dibuat adalah titik awal yang sangat baik untuk mengumpulkan entri kamus data [3].

Salah satu alasan penting untuk memelihara kamus data adalah untuk menjaga kebersihan data. Ini berarti bahwa data harus konsisten. Jika Anda menyimpan data tentang jenis kelamin pria sebagai "M" dalam satu catatan, "Pria" di catatan kedua, dan sebagai nomor "1" dalam catatan ketiga, maka dapat disimpulkan

bahwa data tidak bersih. Menyimpan kamus data akan membantu dalam hal tersebut [3].

Kamus data otomatis (bagian dari alat CASE yang disebutkan sebelumnya) sangat berharga bagi kapasitas mereka untuk referensi silang data, sehingga memungkinkan perubahan program yang diperlukan untuk semua program yang berbagi elemen umum. Fitur ini menggantikan program yang berubah secara tidak teratur, dan mencegah penungguan hingga program tidak berjalan karena perubahan belum diterapkan di seluruh semua program saling berbagi barang yang diperbarui. Jelas, kamus data otomatis penting untuk perusahaan besar dengan sistem yang menghasilkan beberapa ribu elemen data yang memerlukan katalogisasi dan referensi silang [3].

Selain menyediakan dokumentasi dan menghilangkan redundansi, kamus data dapat digunakan untuk [3]:

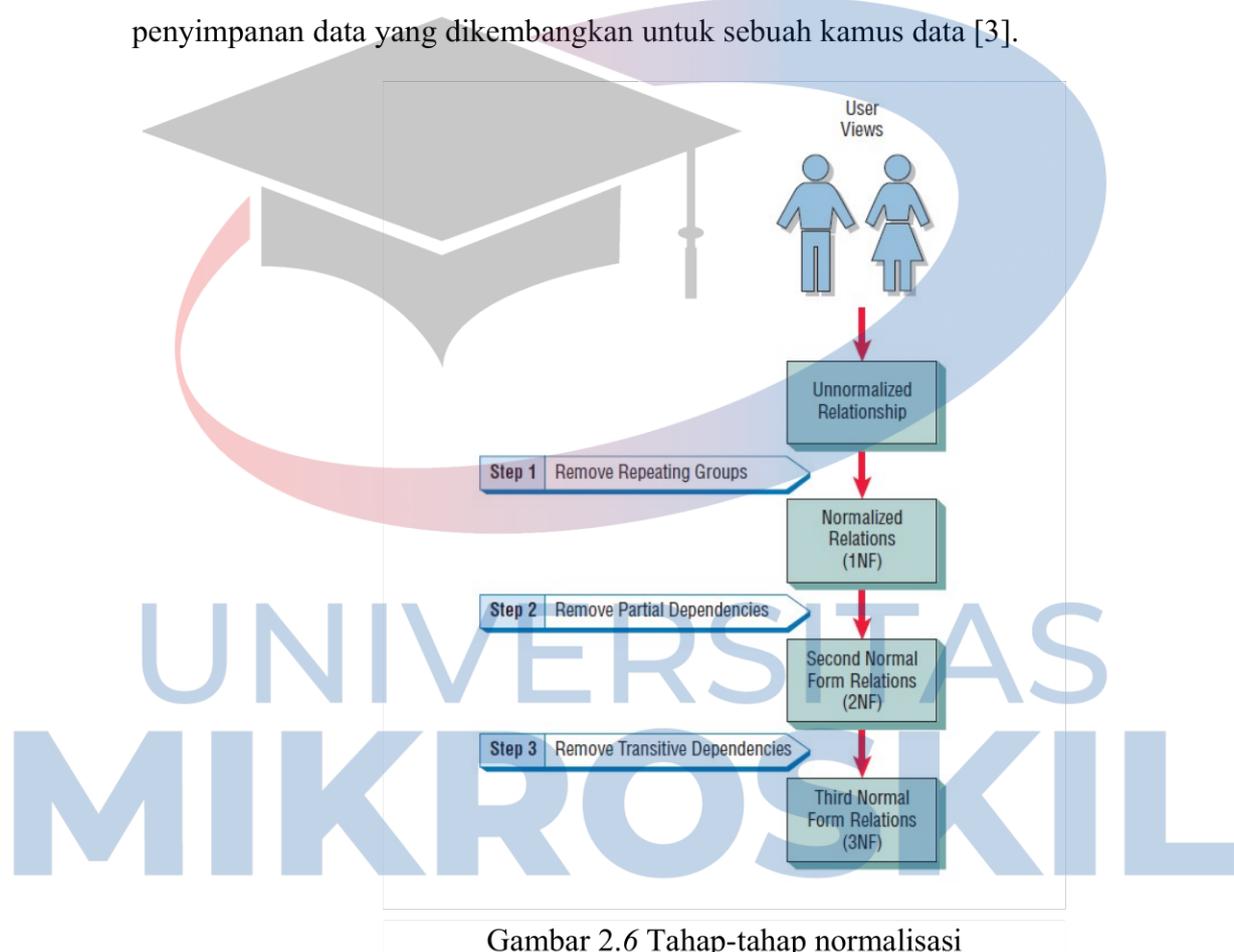
1. Validasi diagram aliran data untuk kelengkapan dan akurasi.
2. Memberikan titik awal untuk mengembangkan layar dan laporan.
3. Menentukan isi data yang disimpan dalam file.
4. Mengembangkan logika untuk proses diagram aliran data.
5. Membuat XML (*Extensible Markup Language*).

Adapun simbol-simbol yang digunakan pada kamus data adalah sebagai berikut [3]:

1. Tanda sama dengan (=) berarti “terdiri dari”.
2. Tanda plus (+) berarti “dan”.
3. Tanda kurung kurawal {} menunjukkan elemen berulang, juga disebut grup atau tabel berulang. Bisa saja terdapat satu elemen berulang atau beberapa dalam satu grup. Grup yang berulang mungkin memiliki kondisi, seperti jumlah pengulangan yang tetap, atau batas atas dan bawah untuk jumlah pengulangan.
4. Tanda kurung [] digunakan untuk situasi ini/atau. Salah satu elemen atau elemen lain mungkin ada, tapi tidak keduanya. Elemen-elemen yang tercantum di antara tanda kurung adalah saling eksklusif.
5. Tanda kurung () digunakan untuk elemen opsional. Elemen opsional dapat dikosongkan pada layar entri dan mungkin berisi spasi atau nol untuk bidang numerik dalam struktur *file*.

2.3.5 Normalisasi

Normalisasi merupakan perubahan dari tampilan pengguna yang kompleks dan penyimpanan data menjadi sekumpulan data yang lebih kecil dan stabil. Struktur data yang dilakukan normalisasi akan lebih mudah dipelihara daripada struktur data lainnya. Dalam normalisasi terdapat tiga tahap yang harus dijalankan seperti yang ditampilkan pada Gambar 2.6. Untuk melakukan tiga tahapan tersebut, seorang analis menormalisasikan sebuah struktur data yang dimulai dengan tampilan pengguna atau penyimpanan data yang dikembangkan untuk sebuah kamus data [3].



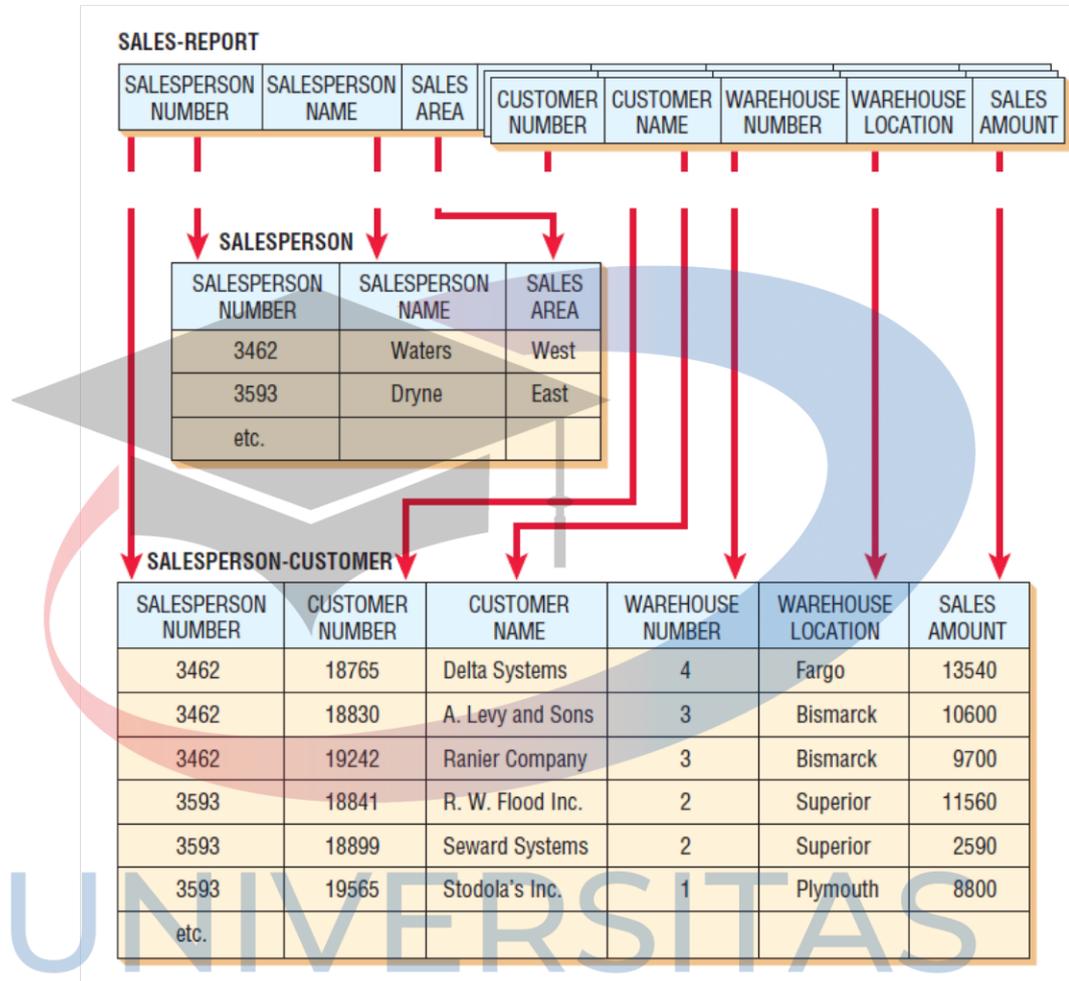
Gambar 2.6 Tahap-tahap normalisasi

Adapun tahap-tahap normalisasi dijelaskan sebagai berikut [3]:

1. Tahap pertama

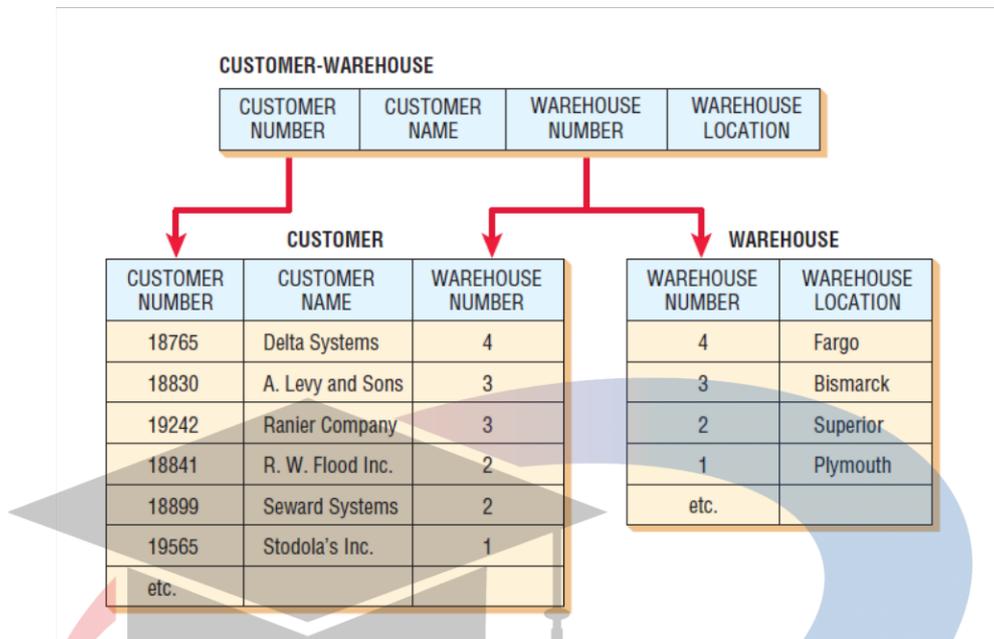
Pada tahap ini dilakukan penghapusan semua kelompok-kelompok yang berulang dan mengidentifikasi *primary key* yang terdapat pada kelompok tersebut. Untuk melakukannya, relasi perlu dipecah menjadi dua atau lebih relasi. Pada titik ini, relasi mungkin telah menjadi bentuk normal ketiga (3NF),

tetapi kemungkinan diperlukan lebih banyak langkah untuk mengubah relasi ke bentuk normal ketiga. Contoh hasil tahap ini dapat dilihat pada Gambar 2.7.

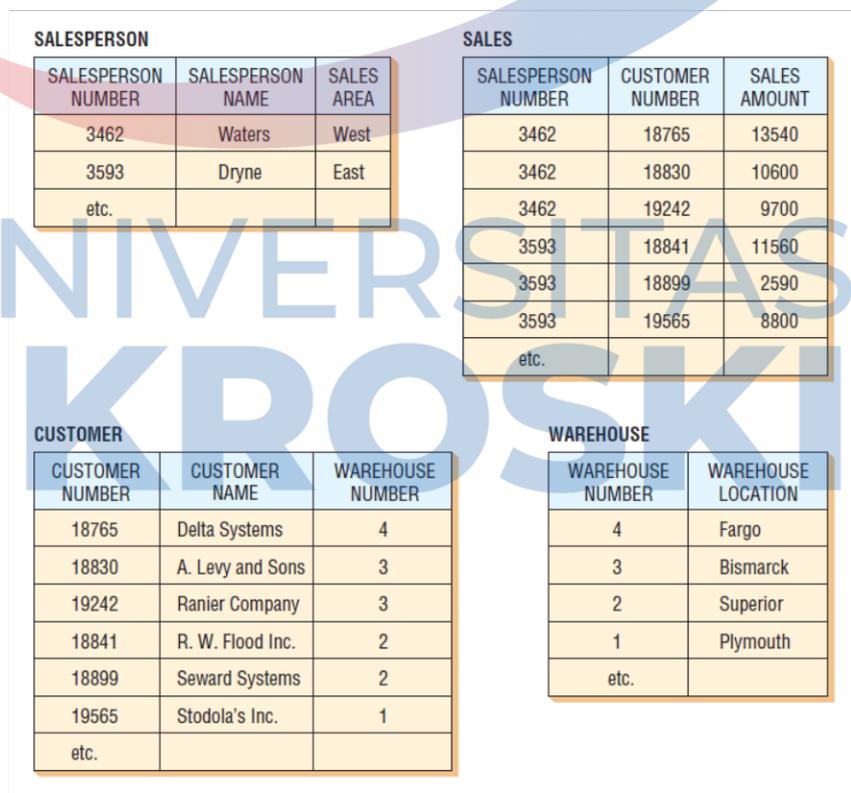


2. Tahap kedua

Tahap ini melibatkan kegiatan memastikan semua atribut bukan kunci sudah bergantung sepenuhnya dengan kunci utamanya (*primary key*). Semua ketergantungan parsial dihapus dan diletakkan pada relasi lainnya. Contoh hasil pada tahap ini seperti yang ditampilkan pada Gambar 2.8.



Gambar 2.9 Tahap ketiga normalisasi



Gambar 2.10 Hasil akhir normalisasi

2.1.

2.2.

2.3.

2.4. Basis Data

Basis data adalah sumber pusat data yang dimaksudkan untuk dibagikan oleh banyak pengguna untuk berbagai aplikasi. Pusat dari basis data adalah *Database Management System* (DBMS), yang memungkinkan pembuatan, modifikasi, dan pembaruan basis data, pengambilan data, dan pembuatan laporan serta tampilan. Administrator basis data adalah orang yang memastikan sebuah basis data memenuhi tujuan.

Tujuan efektivitas dari sebuah basis data adalah sebagai berikut [3]:

1. Memastikan data dapat dibagikan di antara pengguna untuk berbagai aplikasi.
2. Mempertahankan data tetap akurat dan konsisten.
3. Memastikan semua data yang dibutuhkan untuk aplikasi saat ini dan kedepannya akan selalu tersedia.
4. Memungkinkan basis data untuk berkembang seiring dengan peningkatan kebutuhan pengguna.
5. Memungkinkan pengguna untuk membangun pandangan pribadi mengenai data tanpa memperhatikan cara data disimpan secara fisik.

2.5. Perancangan Website

2.5.1 Website

Website atau yang biasa disebut dengan web, merupakan bagian visual dari internet. Web juga dapat diartikan sebagai salah satu aplikasi yang berisikan dokumen-dokumen multimedia berupa teks, gambar, suara, animasi, video, yang di dalamnya mengandung *protocol* HTTP (*hypertext transfer protocol*). Untuk mengakses web tersebut digunakan perangkat lunak yang disebut dengan *browser*.

Berdasarkan aspek isi atau konten, web dapat dibagi menjadi dua jenis, yaitu [7]:

1. *Website* statis, yaitu *website* yang mempunyai isi atau konten yang tidak berubah di mana untuk melakukan perubahan suatu halaman tidak dapat dilakukan secara cepat dan mudah. Hal tersebut dikarenakan teknologi untuk membuat dokumen

web tersebut tidak memungkinkan perubahan isi atau data. Teknologi untuk membuat web statis ini adalah jenis *client side scripting*, seperti HTML, *Cascading Style Sheet (CSS)*.

2. *Website* dinamis, yaitu *website* yang isi atau konten di dalamnya dapat berubah-ubah setiap saat. Untuk membuat jenis web ini, diperlukan beberapa komponen, yaitu *client side scripting* (HTML, JavaScript, CSS), *server side scripting* seperti PHP, dan program basis data seperti MySQL untuk menyimpan data-datanya.

2.5.2 JavaScript

JavaScript merupakan bahasa pemrograman berbentuk kumpulan *script* yang berjalan pada suatu dokumen HTML. JavaScript dapat menyempurnakan tampilan dan sistem pada halaman aplikasi berbasis *website* yang dikembangkan [8].

Bahasa pemrograman JavaScript memiliki karakteristik seperti berikut ini [8]:

1. Bahasa pemrograman yang berjenis *high-level programming*.
2. Bersifat *client-side*.
3. Berorientasi pada objek.
4. Bersifat *loosely typed*.

Aplikasi atau *tools* standar yang digunakan dalam mengimplementasikan JavaScript, yaitu [8]:

1. *Software text-editor*, seperti Notepad++, Adobe Dreamweaver, dan NetBeans.
2. *Web browser*, seperti Mozilla Firefox, Google Chrome, Internet Explorer, OperaMini, Safari, dan lain sebagainya.
3. HTML.

2.5.3 React JS

ReactJS adalah *library* JavaScript yang digunakan untuk mengembangkan komponen antarmuka pengguna (UI) yang dapat digunakan kembali. React pada dasarnya mendukung pengembangan aplikasi berbasis web yang besar dan kompleks, yang dapat mengubah datanya tanpa *refresh* halaman berikutnya. Ini digunakan sebagai tampilan (V) dalam *Model View Controller (MVC)* [9].

React didasarkan pada gagasan bahwa manipulasi DOM adalah operasi yang berat dan harus diminimalkan. React juga mengetahui bahwa mengoptimalkan

manipulasi DOM oleh developer sendiri, akan menghasilkan banyak kode *boilerplate* yang rawan kesalahan, membosankan, dan berulang [10].

React memecahkan masalah ini dengan memberikan *developer* sebuah DOM virtual untuk di-*render* alih-alih menggunakan DOM yang sebenarnya. React menemukan perbedaan antara DOM nyata dan DOM virtual, kemudian melakukan jumlah operasi minimum DOM yang diperlukan untuk menghasilkan *state* baru [10].

React juga bersifat deklaratif. Saat data berubah, React secara konseptual melakukan *refresh* dan tahu untuk hanya memperbarui bagian yang perlu diubah. Aliran data yang sederhana ini, ditambah dengan logika tampilan sederhana, membuat pengembangan dengan ReactJS cepat dan mudah dimengerti [10].

2.5.4 MySQL

MySQL merupakan RDBMS (*server database*) yang mengelola *database* dengan cepat dan menampung dalam jumlah sangat besar, serta dapat diakses oleh banyak pengguna [11]. MySQL adalah sebuah perangkat lunak *open-source* yang digunakan untuk membuat sebuah *database* [12]. Berdasarkan pendapat yang dikemukakan di atas, maka dapat ditarik kesimpulan bahwa MySQL adalah suatu perangkat lunak atau program yang digunakan untuk membuat sebuah basis data yang bersifat *open-source*.

2.1.

2.2.

2.3.

2.4.

2.5.

2.6. Penjualan

Penjualan adalah jumlah yang dibebankan kepada pembeli untuk barang dagang yang diserahkan merupakan pendapat perusahaan yang bersangkutan. Kegiatan penjualan terdiri dari transaksi penjualan barang dan jasa yang dapat dikelompokkan menjadi penjualan kredit dan tunai [13].

Pengertian sistem informasi penjualan, yaitu sub sistem informasi bisnis yang mencakup kumpulan prosedur. Prosedur-prosedur yang disebutkan, yaitu

melaksanakan, mencatat, mengkalkulasi, membuat dokumen dan informasi penjualan untuk keperluan manajemen dan bagian lain yang berkepentingan, mulai dari diterimanya pesanan penjualan hingga mencatat timbulnya tagihan atau piutang dagang [14].

2.7. Pembelian

Pembelian merupakan serangkaian tindakan untuk mendapatkan barang dan jasa melalui penukaran dengan maksud digunakan sendiri atau dijual kembali [15]. Pembelian juga dapat dimaksudkan sebagai suatu sistem kegiatan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan [16].

Dalam sistem pembelian, terdapat dua jenis, yaitu pembelian tunai dan pembelian kredit. Sistem pembelian tunai merupakan sistem yang diberlakukan oleh perusahaan untuk pengadaan barang yang diperlukan perusahaan di mana untuk mendapatkan barang tersebut dilakukan pembayaran lebih dahulu. Sedangkan, sistem pembelian kredit merupakan sistem pembelian di mana pembelian barang dengan pembayaran tempo atau menunda pembayaran atau kredit serta pembayarannya dilakukan setelah barang diterima pembeli. Jumlah dan jatuh tempo pembayarannya disepakati oleh kedua pihak [16].

2.8. Persediaan

Persediaan juga disebut sebagai inventori merupakan sumber daya yang dapat digunakan tetapi tidak / belum dipergunakan. Fungsi pokok dari persediaan adalah memenuhi semua permintaan pelanggan dengan persediaan barang yang seminimal mungkin [17]. Secara rinci, persediaan juga dapat dijelaskan sebagai barang tersedia untuk dijual untuk kegiatan usaha, dalam proses produksi untuk penjualan tersebut, atau dalam bentuk bahan baku atau perlengkapan untuk digunakan dalam proses produksi atau pemberian jasa [18].

Terdapat dua sistem persediaan yang umum untuk melihat jumlah persediaan pada akhir periode, yaitu periodik dan perpetual. Sistem periodik adalah sistem dengan melakukan penghitungan barang secara langsung agar sisa barang dapat diketahui secara pasti pada setiap akhir periode. Sedangkan, sistem perpetual adalah

sistem di mana penghitungan barang dilakukan setiap kali ada transaksi yang dilakukan [18].

2.9. Optik

Istilah optik berasal dari bahasa Latin yang artinya adalah tampilan. Benda optik merupakan benda yang menggunakan lensa optik untuk melakukan fungsinya dalam membantu kegiatan tertentu. Salah satu benda optik yang banyak digunakan adalah kacamata, sebuah lensa tipis untuk menormalkan dan mempertajam penglihatan mata [19].

Optik juga dapat menjadi sebutan untuk perusahaan, toko, atau gerai yang melakukan kegiatan usaha di bidang kacamata. Optik kacamata disebut toko ritel yang menjual langsung produknya kepada konsumen akhir dengan menggunakan pelayanan jasa dalam proses jual beli yang dilakukan untuk membantu konsumen dalam menentukan pilihan [20].

UNIVERSITAS
MIKROSKIL