

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Pengertian Sistem informasi

Sistem merupakan suatu kumpulan objek, komponen, unsur, atau prosedur yang terhubung satu sama lain dan bekerja secara sistematis untuk mencapai tujuan atau maksud tertentu. Informasi adalah data yang memiliki nilai yang akan bermanfaat oleh penggunanya. Data tersebut akan diolah, dibentuk, atau dimanipulasi sesuai dengan keperluan tertentu bagi penggunanya sehingga menghasilkan nilai yang lebih bermakna dan bermanfaat untuk mengambil sebuah keputusan yang konkret. Sistem informasi secara umum dapat diartikan sebagai suatu sistem yang terdiri dari komponen-komponen saling terintegrasi, saling berhubungan dan saling berinteraksi untuk menyediakan informasi yang bermanfaat bagi penggunanya untuk mencapai tujuan tertentu [3].

Sistem informasi juga memiliki komponen-komponen dasar [4]:

1. Perangkat keras komputer (*Hardware*), perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan (*setup*) data, dan terminal masukan/keluaran.
2. Perangkat lunak komputer (*Software*), perangkat lunak umum seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
3. Basis data (*Database*), *file* yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti disket, *hard disk*, *magnetic tape*, dan sebagainya.
4. Jaringan komputer (*Computer Network*), jaringan komputer adalah kumpulan komputer, printer, dan perangkat lainnya terhubung dalam satu kesatuan. Informasi atau data yang ditransmisikan melalui kabel-kabel yang memungkinkan pengguna jaringan komputer untuk bertukar *file* dan data.
5. Prosedur (*Procedure*), prosedur merupakan bentuk fisik seperti buku panduan dan instruksi seperti instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.
6. Pengguna (*Brainware*), pengguna atau orang yang dimaksudkan untuk memakai atau mengoperasikan sebuah perangkat komputer seperti operator komputer, analis sistem, programmer, personil data *entry*, dan manajer sistem informasi.

2.1.2 Komponen Pembangun Sistem Informasi

Komponen-komponen yang membangun sistem informasi dinamakan blok bangunan (*building block*). Blok bangunan ini mirip dengan karakteristik sistem yang telah diuraikan sebelumnya. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan, yang mencakup dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran [3]:

a. Blok masukan (*Input block*)

Masukan mewakili data yang masuk disertai dengan metode dan media yang digunakan untuk menangkap dan memasukkannya ke dalam sistem informasi. Masukan dapat berupa data seperti dokumen-dokumen dasar.

b. Blok model (*Model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan *mathematical models* yang akan memanipulasi data masukan dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

c. Blok keluaran (*Output block*)

Blok ini menghasilkan keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

d. Blok teknologi (*Technology block*)

Blok ini digunakan untuk menerima masukan, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

e. Blok basis data (*Database block*)

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut.

f. Blok kendali (*Control block*)

Blok kendali dirancang untuk meyakinkan bahwa bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.2 Analisis dan Perancangan Sistem Informasi

Analisis dan perancangan sistem informasi digunakan untuk memahami kebutuhan orang-orang dalam menganalisis masukan data atau aliran data secara sistematis, memproses data, menyimpan data, dan menghasilkan keluaran informasi dalam suatu organisasi atau perusahaan. Tujuannya adalah untuk mengidentifikasi dan memecahkan masalah yang tepat melalui analisis sistem klien secara menyeluruh. Analisis dan perancangan sistem informasi digunakan untuk mengimplementasikan peningkatan fungsi bisnis yang ingin dicapai dengan menganalisis, merancang, dan menerapkan perbaikan pada sistem informasi terkomputerisasi yang mendukung pengguna dan kegiatan bisnis [5].

Perancangan sistem informasi terbagi menjadi 2 bentuk yaitu [6][7]:

1. Desain sistem konseptual (*Conceptual system design*)

Proses pengaturan kelayakan dalam penyusunan sistem informasi bergantung pada desain sistem konseptual sebagai pusat atau titik tumpu. Desain konseptual tersebut mencakup penggambaran, perencanaan, dan pengaturan beberapa elemen atau komponen terpisah menjadi satu kesatuan yang berfungsi secara efektif. Dengan demikian, desain konseptual menentukan interaksi lingkaran umpan balik antara materi, energi, dan entitas informasi dalam sistem terbuka. Tahap awal dalam membuat desain konseptual sistem informasi adalah mendefinisikan masalah, menetapkan kendala sistem, menentukan kebutuhan informasi, menetapkan sumber informasi, mengembangkan beberapa alternatif desain konseptual, memilih salah satu desain, mendokumentasikan konsep sistem, dan akhirnya menyiapkan laporan desain konseptual.

2. Desain sistem fisik (*Physical system design*)

Proses desain fisik merupakan tahap selanjutnya dalam pembuatan sistem informasi setelah desain sistem konseptual. Pada tahap ini, analisis akan menerjemahkan persyaratan desain konseptual yang sudah luas dan berorientasi kepada pengguna menjadi spesifikasi detail yang diperlukan untuk mengkode dan menguji program komputer. Selain itu, analisis juga perlu untuk mendesain dokumen input dan output, membuat sejumlah file dan database, mengembangkan prosedur, dan membangun pengendalian yang dibutuhkan ke dalam sistem baru. Setelah tahap desain fisik selesai, hasil dari desain tersebut akan didokumentasikan dan disampaikan kepada panitia pengarah sistem informasi (*information systems steering committee*) sebagai acuan selanjutnya dalam tahap implementasi sistem.

2.2.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah istilah yang umumnya menggambarkan fase awal dari pengembangan sistem. Analisis sistem merupakan faktor penting bagi keberhasilan pengembangan perangkat lunak. Dengan bantuan analisis sistem, organisasi dapat mengetahui masalah apa yang terjadi pada sistem lama [3]. Analisis sistem berfungsi untuk menentukan kebutuhan dari pengguna yang terbagi menjadi dua kebutuhan yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional mencakup setiap proses-proses apa saja atau layanan apa saja yang diimplementasikan dalam sistem, termasuk bagaimana cara sistem harus merepons pada masukan tertentu dan bagaimana perilaku sistem harus beroperasi pada situasi tertentu. Disisi lain, kebutuhan non-fungsional adalah jenis kebutuhan yang mencakup karakteristik perilaku yang dimiliki oleh sistem yang terkait dengan kinerja, operasional, platform sistem dan sebagainya [8]. Adapun beberapa alasan pentingnya mengawali analisis sistem [9]:

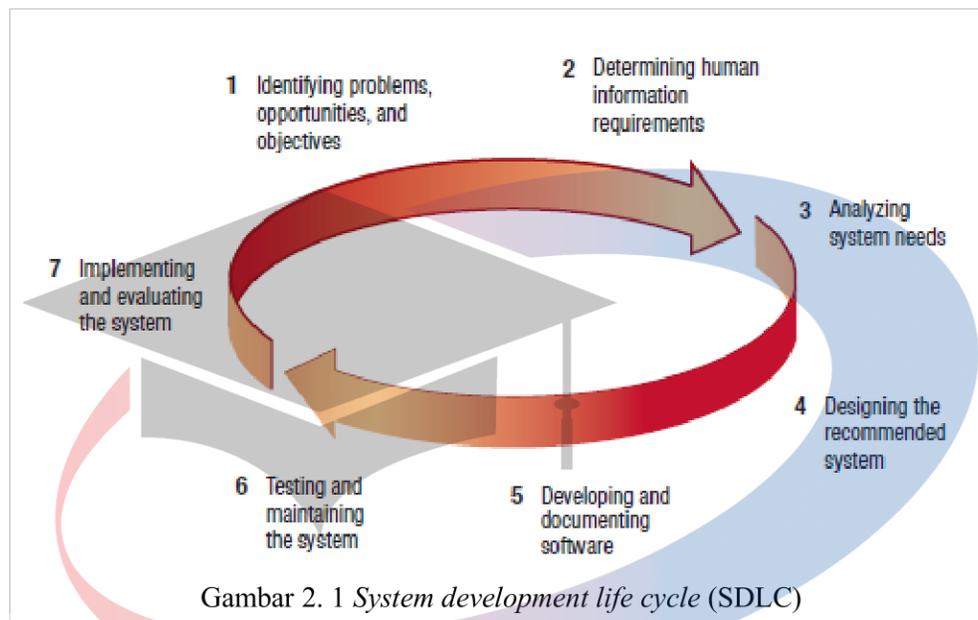
1. Problem-solving: sistem lama yang tidak berfungsi sesuai dengan kebutuhan. Oleh karena itu analisis diperlukan untuk memperbaiki sistem agar dapat berfungsi sesuai dengan kebutuhan.
2. Kebutuhan baru: adanya kebutuhan baru dalam organisasi atau lingkungan sehingga diperlukan adanya modifikasi atau tambahan sistem informasi untuk mendukung organisasi.
3. Mengimplementasikan ide atau teknologi baru: seiring dengan kemajuan teknologi maka sistem juga harus diperbarui agar mampu bersaing dengan kompetitor.
4. Meningkatkan kinerja sistem secara keseluruhan: disini diperlukan agar sistem yang ada saat ini dapat melakukan pengolahan data yang lebih cepat dan menampung data lebih banyak.

2.3 System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) adalah pendekatan secara bertahap guna untuk menganalisis dan mendesain sistem serta memastikan bahwa sistem yang dikembangkan melalui

penggunaan siklus atau langkah-langkah khusus yang disesuaikan dari kebutuhan pengguna dan juga sistem secara spesifik [5].

Setiap tahapan dalam SDLC memiliki karakteristiknya tersendiri yang saling berhubungan. Tahapan dalam SDLC terbagi menjadi 7 tahapan, yaitu [10][11]:



Gambar 2. 1 *System development life cycle (SDLC)*

1. Identifikasi Masalah, Peluang, dan Tujuan

Pada tahap pertama, analis harus memperhatikan dengan teliti masalah apa yang dihadapi dan perlu diidentifikasi, peluang, serta tujuannya. Tahapan ini juga mengurangi duplikasi dan pekerjaan yang sia-sia serta menciptakan suatu pengembangan sistem baru yang sejalan dengan rencana strategis organisasi. Analis dapat melakukan wawancara dan observasi lapangan untuk menentukan kebutuhan pengguna dan sistem. Hasil dari tahap ini adalah laporan kelayakan yang berisi definisi masalah dan merangkum tujuan. Tahapan pertama dapat dilakukan menggunakan diagram *fishbone* untuk mengidentifikasi dan merangkum masalah yang ada lalu untuk mengidentifikasi peluang dan tujuan bisa melalui observasi lapangan dan wawancara kepada perusahaan atau organisasi.

2. Menentukan persyaratan informasi manusia

Pada tahap kedua, analis akan melakukan observasi identifikasi kebutuhan pengguna sistem dan memahami bagaimana pengguna akan berinteraksi dengan sistem yang telah ada sebelumnya. Analis akan melakukan tahap mengidentifikasi kebutuhan informasi, menganalisis kebutuhan informasi pengguna yang terlibat dalam sistem dan proses dalam sistem serta orang-orang yang

terlibat di dalamnya. Tahapan kedua adalah proses menggambar struktur perusahaan, menentukan persyaratan yang diperlukan untuk membuat sistem yang akan digunakan dan terakhir menganalisis dokumen masukan dan keluaran.

3. Analisis kebutuhan sistem

Pada Tahap selanjutnya, analis akan melakukan analisis kebutuhan sistem dengan memetakan kebutuhan sistem untuk menggambarkan masukan dan keluaran yang dibutuhkan sistem. Dalam tahap ini terdapat *tools* dan teknik untuk mendukung penentuan kebutuhan, salah satunya adalah *Data Flow Diagram* (DFD) yang dapat memetakan *input*, *process*, dan *output* dari suatu fungsi bisnis. Penggunaan DFD juga sebagai acuan informasi dan lokasi dimana informasi yang keluar dan masuk digambarkan dengan jelas. Hasil dari tahap ini adalah penyusunan proposal sistem yang akan meringkas informasi apa saja yang telah ditemukan tentang pengguna, kegunaan, dan kegunaan sistem saat ini. Pada tahapan ketiga, proses yang dikerjakan pada tahapan ini adalah membuat *data flow diagram* (DFD) sistem berjalan dan sistem usulan yang akan dirancang serta mengidentifikasi kebutuhan fungsional dan non-fungsional dengan menggunakan metode PIECES.

4. Merancang sistem yang direkomendasikan

Tahap ini bertujuan untuk merancang sistem menggunakan informasi yang telah dikumpulkan sebelumnya untuk membuat desain logis dari sistem informasi. Analis merancang prosedur *data entry* bagi pengguna agar data dapat dimasukan secara akurat sehingga dapat dikatakan *valid*. Selain itu, analis akan menyediakan antarmuka pengguna (*User interface*) untuk memastikan bahwa sistem aman, dapat dibaca, dan mudah untuk digunakan. Tahap desain juga akan merancang *database* untuk menyimpan sejumlah besar data yang dibutuhkan oleh para pengambil keputusan dalam organisasi. Di tahapan keempat, proses yang pertama adalah merancang format laporan sebagai keluaran (*output*) sistem usulan dengan aplikasi Crystal Report lalu merancang bentuk antarmuka pemakai (*user interface*) dari masukan (*input*) sistem usulan dengan Microsoft Visual Studio serta merancang basis data (*database*) yang digunakan oleh sistem usulan yang terdiri dari struktur tabel menggunakan Microsoft SQL Server.

5. Mengembangkan dan mendokumentasikan perangkat lunak

Dalam tahap ini, seorang analis bekerja sama dengan pengguna dalam mengembangkan dokumentasi terhadap perangkat lunak yang efektif, dimana mencakup beberapa hal seperti prosedur manual, bantuan online, serta *website* dengan fitur *Frequently Asked Questions* (FAQ),

di bagian “*Read Me*” yang kemudian dikirimkan bersama-sama dengan perangkat lunak baru. Kegiatan dokumentasi menunjukkan kepada pengguna tentang cara menggunakan perangkat lunak dan apa saja yang harus dilakukan bila perangkat lunak mengalami masalah. Untuk memastikan kualitasnya, pemogram bisa membuat perancangan serta kode program apa saja yang akan dijalankan, kemudian menjelaskan bagian-bagian kompleks dari program tersebut kepada tim pemogram lainnya.

6. Menguji dan mempertahankan sistem

Sebelum sistem informasi dapat digunakan, perlu dilakukan pengujian terlebih dahulu. Hal ini bertujuan untuk menghemat biaya sehingga jika adanya masalah sebelum sistem tersebut diterapkan maka sistem dapat diperbaiki kembali. Sebagian pengujian biasanya dilakukan oleh pemogram sendiri, dan juga oleh penganalisis sistem. Rangkaian pengujian ini dijalankan bersama-sama dengan data sampel serta dengan data aktual dari sistem yang telah ada. Kegiatan uji coba *software* dapat menggunakan “komponen uji kelayakan *prototype*” meliputi komponen: rancangan Input, rancangan Sistem, rancangan *database*, rancangan kendali, rancangan platform teknologi.

7. Mengimplementasi dan mengevaluasi sistem

Implementasi sistem dapat dilakukan setelah rancangan terhadap sistem telah memasuki tahap penyelesaian. Analisis perlu merencanakan konversi yang lancar dari sistem yang lama ke sistem yang baru yang dimana dalam proses ini termasuk di dalamnya adalah mengkonversi *file* dari format lama ke yang baru, membangun *database*, dan penginstalan peralatan instalasi serta penggantian ke sistem yang baru. Pada perangkat keras yang tersedia, dipastikan bahwa perangkat tersebut sudah terpasang dengan baik dan dibuat basis datanya. Kemudian melakukan pelatihan kepada pengguna *software* baru terhadap tata cara penggunaan *software* sebagai kegiatan berikutnya serta melakukan adaptasi terhadap sistem yang baru. Evaluasi dilakukan untuk melihat keberhasilan dalam kegiatan, termasuk didalamnya efektivitas sistem baru, pencocokan terhadap biaya yang dikeluarkan berdasarkan pada perkiraan biaya, kemudian ketepatan waktu dalam pelaksanaan proyek, serta biaya pemeliharaan sistem. Evaluasi diharapkan agar sistem baru lebih efisien dalam operasionalnya dan efektif dalam mencapai tujuan yang dituju, lebih mudah digunakan dan fleksibel terhadap perubahan-perubahan yang diinginkan. Pemeliharaan adalah proses yang dilakukan selama siklus hidup sistem informasi berjalan. Pemeliharaan dilakukan setelah sistem diimplementasi, biasanya dengan cara

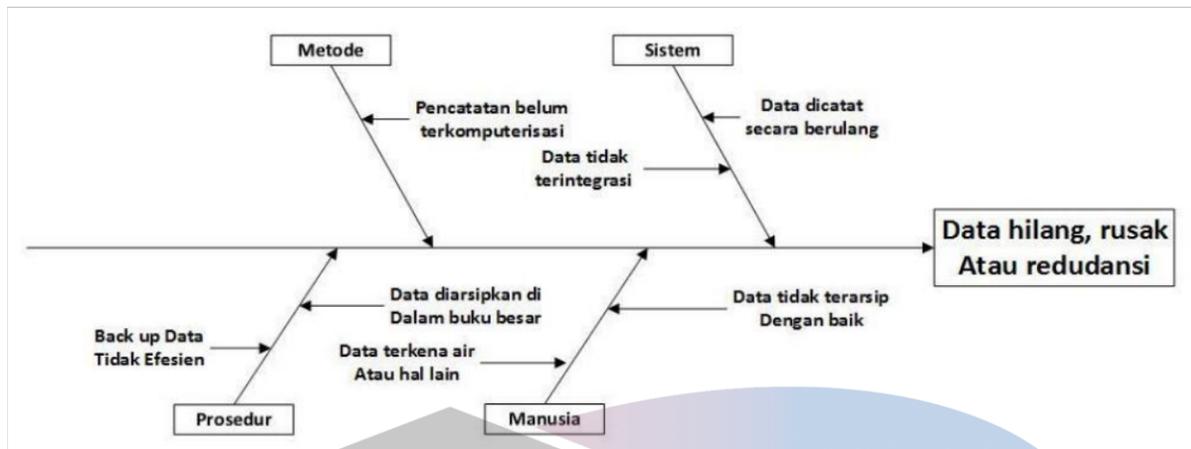
memperbaiki kesalahan program yang tidak terdeteksi sebelumnya. Oleh karena itu, setelah sistem terpasang, maka sistem tersebut harus dipantau lebih lanjut lagi agar program komputer yang sudah diimplementasikan dapat dilakukan modifikasi untuk penyesuaian terhadap kesalahan yang ditemukan dan dijaga supaya tetap terpakai.

2.4 Teknik Pengembangan Sistem

2.4.1 Diagram Fishbone

Diagram *fishbone*, juga dikenal sebagai diagram sebab-akibat, pertama kali digunakan pada tahun 1940 oleh Profesor Kaoru Ishikawa dari Universitas Tokyo untuk memecahkan masalah kualitas. Sebagai hasil karyanya, diagram ini terkadang disebut sebagai diagram Ishikawa. Seiring berjalannya waktu, metode ini telah diterapkan pada hampir semua masalah. Diagram Penyebab & Efek (*Cause & Effect*) ini bertujuan untuk menyelesaikan penyebab masalah dalam tim. Kendala yang muncul saat melakukan proses pemecahan masalah dalam tim adalah adanya perbedaan ide dan persepsi dari masing-masing anggota tim. Diagram *fishbone* membantu memvisualisasikan akar penyebab masalah, sehingga memudahkan tim untuk mengidentifikasi akar penyebab masalah [12]. Adapun beberapa kategori-kategori yang menjadi penyebab masalah diantaranya [13]:

1. Kategori 8P yang biasa digunakan dalam industri jasa yaitu *Product* (produk/jasa), *Price* (harga), *Place* (tempat), *Promotion* (promosi atau hiburan), *People* (orang), *Process* (proses), *Physical Evidence* (bukti fisik), dan *Productivity & Quality* (produktivitas dan kualitas)
2. Kategori 5S yang biasa digunakan dalam industri jasa yaitu *Surroundings* (lingkungan), *Supplier* (pemasok), *System* (sistem), *Skills* (keterampilan), dan *Safety* (keselamatan).
3. Kategori 6M yang biasa digunakan dalam industri manufaktur yaitu *Machine* (mesin atau teknologi), *Method* (metode atau proses), *Material* (termasuk raw material, consumption, dan informasi), *Man Power* (tenaga kerja atau pekerjaan fisik) / *Mind Power* (pekerjaan pikiran: *kaizen*, saran dan sebagainya), *Measurement* (pengukuran atau inspeksi), dan *Milieu / Mother Nature* (lingkungan)



Gambar 2.2 Contoh diagram Fishbone

2.4.2 Analisis PIECES

Metode analisis PIECES adalah kerangka kerja untuk mengklasifikasi persoalan atau permasalahan berdasarkan kategori *performance, information, economic, control, efficiency, service*. Analisis PIECES dapat menemukan beberapa kekurangan dan permasalahan yang ada pada sistem lama untuk menentukan permasalahan utamanya. Dalam menganalisis sebuah sistem informasi, analisis PIECES harus menyoroti enam aspek penilaian, yang diantaranya adalah [14] [15]:

1. Keandalan (*performance*)

Keahlian atau kehandalan dari sebuah sistem informasi adalah hal pertama yang dinilai dari metode PIECES, dimana *performance* bermain peran penting dalam menentukan seberapa handal sistem tersebut dalam memproses atau mengolah data untuk menghasilkan informasi yang memenuhi tujuan yang diharapkan. Analisis ini dilakukan untuk mengetahui kinerja sistem dan mengukur kehandalan dari jumlah temuan data yang dihasilkan dan kecepatan dalam menemukan data.

2. Informasi (*information*)

Informasi sangatlah penting dan dapat membantu pihak manajemen dan user untuk melakukan pengambilan keputusan yang sesuai, maka dari itu kehandalan dari sebuah sistem informasi memainkan peran yang sangat penting karena jika sistem bekerja dengan baik, maka user akan menerima informasi yang akurat, tepat waktu dan relevan sesuai dengan harapan mereka. Analisis ini juga dilakukan untuk mengetahui seberapa banyak serta seberapa akurat informasi yang akan dihasilkan oleh suatu pencarian.

3. Nilai ekonomis (*economic*)

Ekonomis dalam sistem informasi menekankan pada pemanfaatan biaya yang efektif dalam menggunakan informasi. Analisis ekonomi digunakan untuk memastikan bahwa penerapan sistem informasi telah memenuhi kriteria efisiensi finansial dan mengurangi biaya, serta memperoleh manfaat maksimal bagi lembaga informasi. Penggunaan dari kebutuhan informasi yang ekonomis dapat mempengaruhi pengendalian biaya dan meningkatkan manfaat pada sebuah sistem informasi

4. Analisis pengendalian (*control*)

Sebaik-baiknya suatu sistem, jika tidak dilengkapi dengan pengendalian dan pengamanan yang baik, dapat menjadi lemah dan mudah untuk dirusak oleh pihak luar. Oleh karena itu, pengendalian dan pengamanan sistem informasi sangat penting untuk dipantau dan dilakukan agar sistem dapat berjalan dengan baik. Analisis ini juga dilakukan untuk mengevaluasi sejauh mana pengawasan dan kontrol yang dilakukan untuk memastikan kinerja sistem informasi yang optimal dari segi integritas sistem, kemudahan akses, dan keamanan data.

5. Analisis efisiensi (*efficiency*)

Sistem informasi yang digunakan harus memiliki keunggulan jika dibandingkan dengan penggunaan sistem secara manual, yaitu dengan tingkat keefisienan yang baik dalam operasional. Efisiensi dan efektivitas sistem perlu dipertanyakan dalam kinerja dan alasan pembuatan sistem tersebut. Analisis dilakukan untuk menentukan apakah suatu sistem efisien, dengan sedikit input yang dapat menghasilkan output yang memuaskan dan membantu menyelesaikan permasalahan secara otomatis.

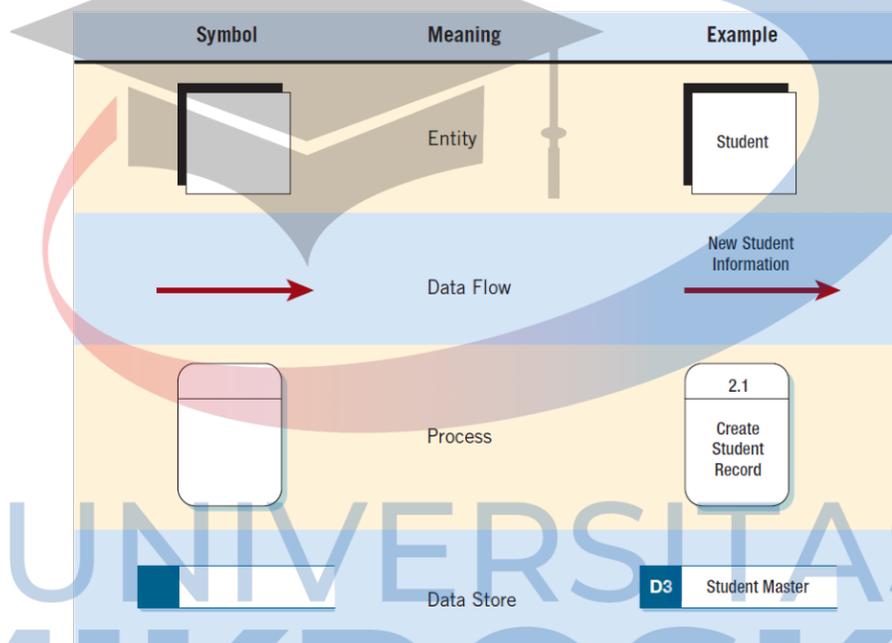
6. Pelayanan (*Service*)

Pelayanan kepada konsumen merupakan faktor penting yang harus diperhatikan dalam sistem informasi. Pelayanan yang baik akan mempengaruhi kepuasan pengguna sistem dan meminimalisir kemungkinan mereka beralih ke pesaing lain. Analisis dalam hal ini digunakan untuk mengukur tingkat pelayanan yang diterima pengguna sistem dan mengidentifikasi permasalahan terkait pelayanan serta dapat memberikan solusi dari permasalahan terkait.

2.4.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah model logika data atau proses yang digunakan untuk menggambarkan dari mana data berasal dan kemana tujuan data yang keluar dari sistem, dimana

data disimpan, proses mana yang menghasilkan data, dan interaksi antara data yang disimpan, dan proses yang dikenakan pada data tersebut. DFD juga merupakan alat untuk mengvisualisasikan sistem yang kompleks dengan membaginya menjadi modul-modul yang lebih kecil agar membuat sistem lebih detail dan mudah dipahami [16]. DFD terdiri dari diagram konteks dan diagram rinci (DFD *Levelled*). Diagram konteks digunakan untuk memetakan model lingkungan (menggambarkan hubungan antara entitas eksternal, *input*, dan *output* sistem), dan digambarkan oleh satu lingkaran tunggal yang mewakili keseluruhan sistem. DFD *levelled* menggambarkan sistem sebagai jaringan antar fungsi yang terhubung satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari perspektif fungsional [17].



Gambar 2. 3 Simbol DFD (*Gane & Sarson DFD Symbols*)

Terdapat penjelasan mengenai simbol dari data flow diagram adalah sebagai berikut [5]:

1. Entitas Eksternal (*External Entity*)

Entitas eksternal mewakili entitas apa pun yang menyediakan atau menerima informasi dari sistem, tetapi bukan bagian dari sistem itu sendiri. Entitas ini bisa berupa individu, sekelompok orang, jabatan atau departemen perusahaan, atau sistem lainnya.

2. Aliran Data (*Data Flow*)

Panah menunjukkan pergerakan data dari satu titik ke titik lain, dengan ujung panah mengarah ke tujuan data tersebut. Aliran data yang terjadi secara bersamaan dapat digambarkan melalui

penggunaan panah sejajar. Karena panah mewakili data tentang orang, tempat, atau hal, panah juga sebaiknya dijelaskan dengan kata benda.

3. Proses (*Process*)

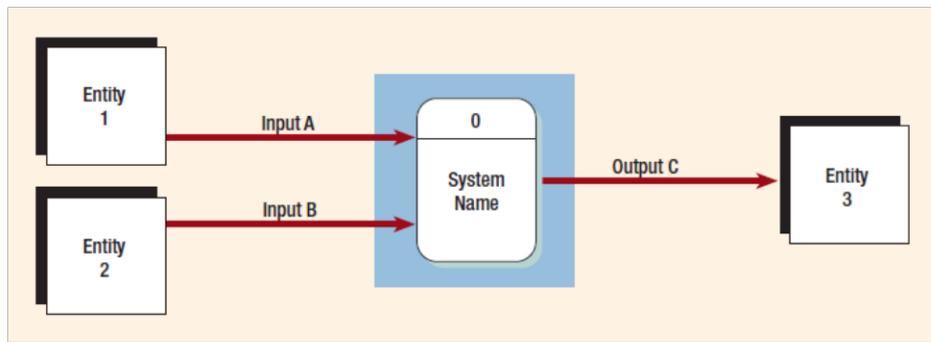
Sebuah persegi panjang dengan sudut melengkung digunakan untuk menunjukkan terjadinya proses transformasi. Proses selalu mengindikasikan perubahan atau transformasi data; oleh karena itu, aliran data yang keluar dari proses selalu diberi label yang berbeda dari yang masuk ke dalamnya. Proses mewakili pekerjaan yang dilakukan dalam sistem dan harus diberi nama dengan salah satu format berikut. Sebuah proses juga harus diberi nomor identifikasi unik yang menunjukkan tingkatnya dalam diagram.

4. Penyimpanan Data (*Data Store*)

Simbol dasar terakhir yang digunakan dalam DFD adalah persegi panjang dengan ujung terbuka, yang mewakili penyimpanan data. Pada DFD logis, jenis penyimpanan fisik tidak ditentukan. Pada tahap ini, simbol penyimpanan data hanya menunjukkan tempat penyimpanan data yang memungkinkan pemeriksaan, penambahan, dan pengambilan data. Penyimpanan data dapat mewakili penyimpanan manual, seperti lemari arsip, atau file atau basis data terkomputerisasi. Karena penyimpanan data mewakili orang, tempat, atau hal, mereka diberi nama dengan kata benda. Penyimpanan data sementara, seperti kertas coretan atau file komputer sementara, tidak termasuk dalam DFD. Berikan setiap penyimpanan data nomor referensi unik, seperti D1, D2, D3, dan seterusnya.

2.4.4 Diagram Konteks

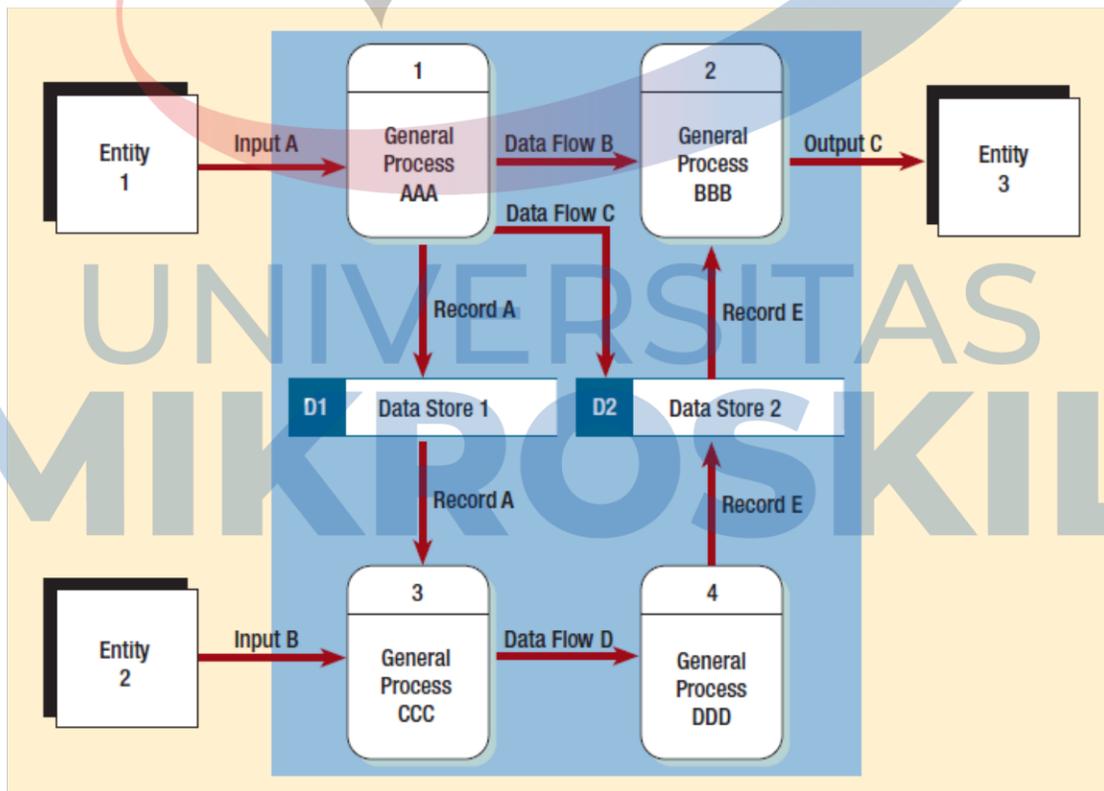
Diagram konteks adalah diagram yang terdiri dari proses yang menggambarkan ruang lingkup suatu proses sistem. Diagram konteks adalah *level* tertinggi dari DFD, yang menggambarkan semua *input* ke sistem atau *output* dari sistem. Semua entitas eksternal harus digambarkan dengan jelas sehingga aliran data dapat dibentuk melalui *input*, proses dan *output*. Diagram konteks hanya terdiri dari satu proses saja dan umumnya tidak ada data store di dalam konteks diagram. Diagram konteks berisi siapa saja (entitas) yang menyediakan data ke sistem serta kepada siapa saja informasi yang harus dihasilkan sistem [17].



Gambar 2. 4 Diagram Konteks

2.4.5 Data Flow Diagram Nol (Diagram overview)

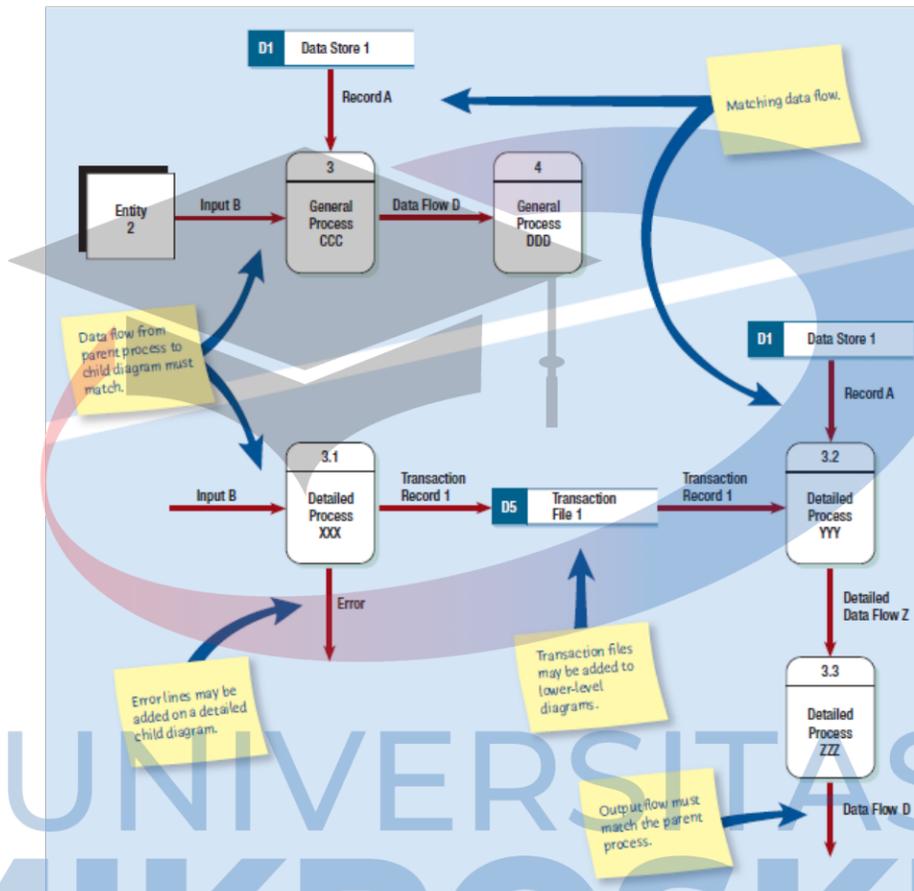
Diagram *overview* atau DFD *level 0* adalah tahapan atau pemecahan dari proses utama diagram konteks menjadi beberapa proses penting yang lebih mudah dipahami sesuai dengan perbedaan prosesnya. Diagram 0 memberikan gambaran umum tentang sistem yang telah dijabarkan di diagram konteks lalu menguraikan ke subproses yang lebih rinci lagi [18].



Gambar 2. 5 DFD level 0

2.4.6 Data Flow Diagram Rinci (DFD Level Anak)

Diagram rinci atau diagram *level 1* menggambarkan proses yang lebih mendetail dari proses utama yang telah diuraikan pada DFD level 0. Proses utama yang sudah di jabarkan pada Diagram overview diuraikan kembali menjadi beberapa proses yang lebih rinci lagi hingga pada tahap tertentu dimana sudah tidak dapat diuraikan lagi [18].



Gambar 2. 6 Diagram Level 0 (atas) Diagram Anak (bawah)

2.5 Kamus Data

Kamus data adalah kumpulan daftar elemen data yang mengalir melalui sistem perangkat lunak sehingga *input* dan *output* dapat dipahami secara umum. Kamus data adalah suatu elemen yang beraturan dengan definisi tetap dan sesuai dengan sistem, sehingga pengguna dan analis sistem memahami komponen *input*, *output*, dan gudang data dengan cara yang sama. Kamus data dibuat dalam fase analisis sistem dan akan digunakan dalam fase analisis maupun fase desain sistem. Pada tahap analisis, kamus data digunakan sebagai alat komunikasi antara analis sistem dan pengguna sistem tentang informasi yang mengalir dalam sistem, yaitu data yang masuk ke sistem

dan informasi yang dibutuhkan oleh pengguna sistem. Selama fase desain sistem, kamus data digunakan untuk merancang *input*, laporan, dan *database* [19]. Selain menyediakan dokumentasi dan menghapuskan redundansi, kamus data dapat digunakan [5]:

1. Melakukan validasi pada diagram aliran data untuk memeriksa kelengkapannya dan keakuratannya.
2. Menjadikan sebagai titik awal untuk mengembangkan tampilan dan laporan.
3. Menentukan jenis data yang akan disimpan dalam file.
4. Membangun logika untuk memproses diagram aliran data.
5. Buat bahasa markup yang dapat diperluas dengan format XML.

Tabel 2. 1 Simbol-simbol Kamus Data

No.	Simbol	Keterangan
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 <i>field-field</i> numerik pada struktur <i>file</i> .
4.	{ }	Menunjukkan elemen-elemen <i>repetitive</i> , juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa 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 satu sama lain. (Dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)

6.		Pemisah sejumlah alternatif pilihan antara simbol []
7.	@	Identifikasi atribut kunci
8.	**	Batas komentar



Gambar 2. 7 Contoh Kamus data

2.6 Konsep Basis Data

2.6.1 Pengertian Basis Data

Basis data terdiri dari dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai pangkalan ataupun gudang, suatu tempat untuk berkumpul, sedangkan data merupakan representasi dari informasi dunia nyata yang mewaliki suatu objek. Bahasa basis data biasanya dapat dipilih dalam dua bentuk yaitu: “*Data Defenition Language (DDL)* dan *Data Manipulation Language (DML)* [20].

Data Definition Language (DDL) digunakan untuk mengelola obyek *database* seperti *database*, *tables*, dan *views*. DDL memiliki fungsi utama untuk mendefinisikan data dalam *database* secara logika, diantaranya adalah [20]:

1. Untuk menentukan properti dari *record* (meliputi nama, tipe, dan lebar dari field).
2. Untuk memutuskan *key field*.
3. Menyediakan cara untuk mendefinisikan hubungan data di field lain.
4. Mengubah struktur dari *record*.
5. Menampilkan struktur dari *record*.

Data Manipulation Language (DML) adalah bentuk bahasa basis data yang berguna untuk memanipulasi data yang terdapat dalam tabel. Data dapat diolah dengan menambah atau menambahkan data baru ke *database*, menghapus dan mengubah data dari *database*. Ada dua jenis DML, yaitu [20]:

1. *Procedural*, yaitu dimana pengguna harus memutuskan informasi apa yang diinginkan dan bagaimana cara untuk mendapatkannya
2. *Non-procedural*, membuat pengguna dapat menentukan data apa yang diinginkan tanpa kesulitan mengetahui bagaimana cara mendapatkannya.

2.6.2 Database Management Systems (DBMS)

DBMS (*Database management system*) adalah sistem pengorganisasian data pada komputer yang memungkinkan pengguna untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. DBMS (*Database management system*) ini adalah perangkat lunak yang dapat digunakan untuk membangun database komputer. Peran dari DBMS merupakan perantara antara pengguna dan database, sehingga pengguna dapat dengan mudah mencari dan menambahkan informasi atau data ke *database* melalui DBMS [20].

Untuk mengakses DBMS, pengguna harus menggunakan bahasa basis data, bahasa basis data adalah sekumpulan pernyataan yang digabungkan bersama untuk diproses oleh DBMS. Perintah atau query biasanya ditentukan oleh pengguna. Adapun contoh-contoh aplikasi perangkat lunak seperti: MySQL, Oracle, dan Microsoft SQL Server [20].

Ada beberapa Komponen Database Management System, antara lain [20]:

1. *File manager* berperan untuk mengelola struktur atau susunan data yang digunakan untuk menampilkan informasi yang ada dalam penyimpanan *disk*.

2. *Database manager* berfungsi adalah menyediakan interface antar data dengan aplikasi dan *query*.
3. *Query Processor* bekerja dalam menerjemahkan perintah dalam bahasa kueri menjadi instruksi tingkat rendah (*Low-level*) yang dipahami oleh *database manager*.
4. *DML Precompiler* adalah komponen yang mengkonversi perintah DML yang ditambahkan dalam suatu program aplikasi ke pemain prosedur normal dalam bahasa induk.
5. *DDL Compiler* digunakan untuk mengubah berbagai perintah DDL menjadi satu set tabel yang berisi metadata.

2.6.3 Normalisasi Data

Normalisasi merupakan transformasi tampilan pengguna yang kompleks dan penyimpanan data menjadi sekumpulan struktur data yang lebih kecil dan stabil. Selain lebih sederhana dan lebih stabil, struktur data yang dinormalisasi lebih mudah dipelihara daripada struktur data lainnya. Normalisasi dimulai dengan penyimpanan data yang dikembangkan untuk kamus data lalu seorang analis akan menguraikannya ke struktur data dalam 3 langkah yaitu 1NF (*first normal form*), 2NF (*second normal form*), 3NF (*third normal form*). Setiap langkah melibatkan proses penting untuk menyederhanakan struktur data. Relasi yang bersumber dari *user view* atau penyimpanan data cenderung tidak dinormalisasikan. Pada langkah pertama dari proses normalisasi melibatkan penghapusan semua atribut data yang duplikat dan menentukan *primary key*. Langkah kedua harus memastikan bahwa semua atribut non-kunci sepenuhnya bergantung pada *primary key*. Semua dependensi persial dihapus dan ditempatkan di relasi yang lain. Langkah ketiga melibatkan penghapusan semua dependensi transitif. Ketergantungan transitif adalah ketergantungan di mana satu atribut non-kunci bergantung pada atribut non-kunci lainnya [5].

1. Bentuk tidak normal (*Unnormalization Form*)

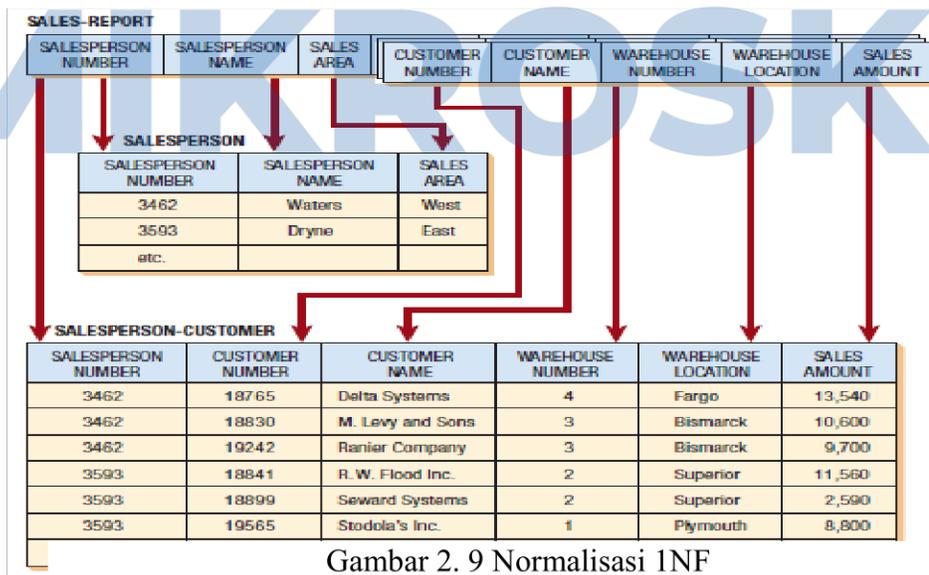
Unnormalization form atau bentuk tidak normal (UNF) dari suatu tabel *database* mengacu pada kumpulan data mentah yang dimasukkan pada satu tabel yang sama (tidak dipecah ke tabel lain). Data-data tersebut masih terdapat pengulangan atau duplikasi (*repeating groups*) yang mungkin tidak sempurna atau tidak lengkap sehingga terjadi permasalahan saat melakukan manipulasi data (*insert, update, dan delete*) atau bisa disebut sebagai anomali [21].

SALESPERSON NUMBER	SALESPERSON NAME	SALES AREA	CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES AMOUNT
3462	Waters	West	18765	Delta Systems	4	Fargo	13,540
			18830	M. Levy and Sons	3	Bismarck	10,600
			19242	Ranier Company	3	Bismarck	9,700
3593	Dryne	East	18841	R. W. Flood Inc.	2	Superior	11,560
			18899	Seward Systems	2	Superior	2,590
			19565	Stodola's Inc.	1	Plymouth	8,800
etc.							

Gambar 2. 8 Unnormalized form UNF

2. Bentuk normal pertama (*First Normal Form*)

Bentuk normal yang pertama atau 1NF memerlukan beberapa kondisi untuk mengelompokkan beberapa jenis data yang sejenis untuk mengatasi masalah anomali. Bentuk normal pertama berfungsi untuk menghilangkan duplikasi data didalam kolom dari tabel yang sama

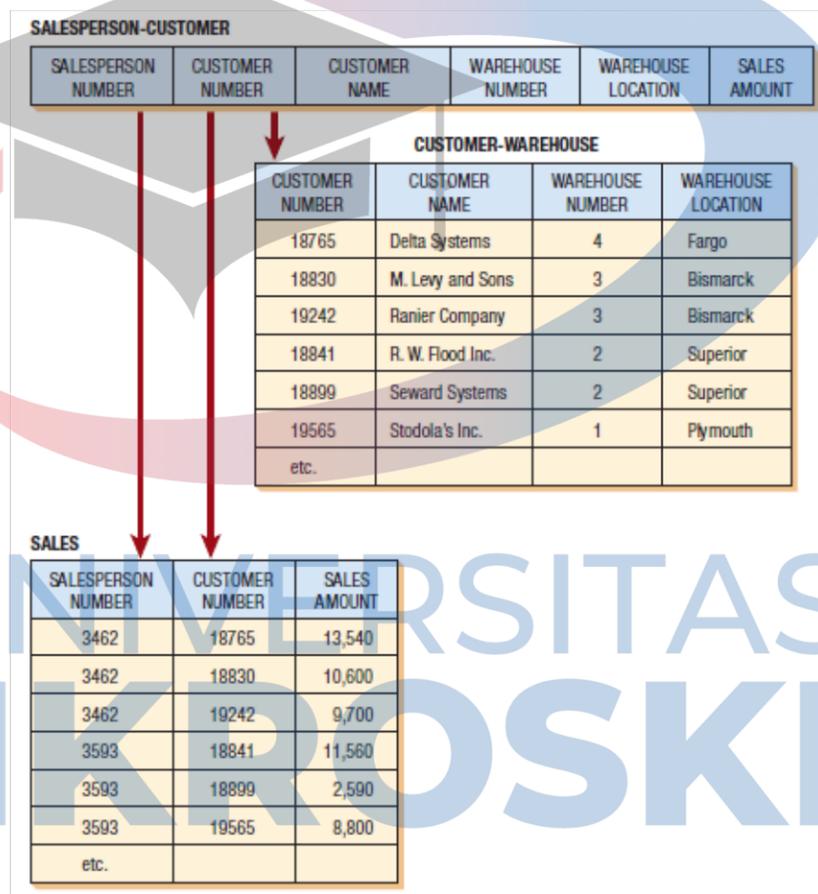


Gambar 2. 9 Normalisasi 1NF

dan membuat tabel terpisah untuk setiap grup data yang terkait serta mengidentifikasi setiap baris dengan kolom yang unik (*primary key*) [22].

3. Bentuk normal kedua (*Second Normal Form*)

Bentuk normal kedua atau 2NF akan terpenuhi jika data yang ada pada 1NF telah dibuat. Adapun beberapa fungsi dari normalisasi 2NF, seperti menghapus beberapa subset data yang sudah ada di dalam tabel dan meletakkannya di tabel yang terpisah lalu membuat relasi antara tabel baru dan tabel lama dengan membuat *foreign key*. Tidak ada atribut dalam tabel yang secara fungsional bergantung pada *candidate key* tabel [22].



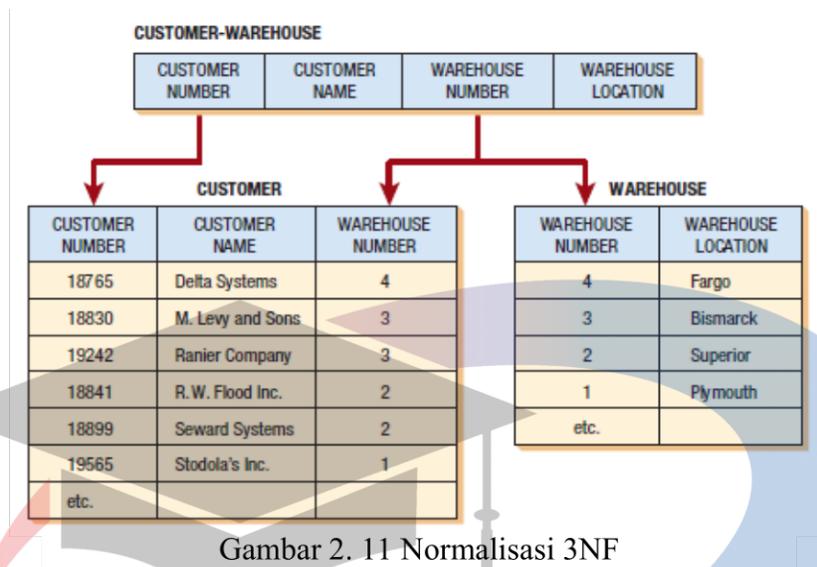
Gambar 2. 10 Normaliasi 2NF

4. Bentuk normal ketiga (*Third Normal Form*)

Bentuk normal ketiga atau 3NF bertujuan untuk menghilangkan semua atribut atau field tabel yang tidak berhubungan dengan *primary* sehingga atribut itu harus dipisah ke tabel yang baru.

Oleh karena itu, setiap *candidate key* tidak memiliki dependensi transitif. Persyaratan bentuk

normal ketiga adalah memenuhi semua persyaratan dari 2NF dan 1NF serta menghapus kolom tabel yang tidak tergantung pada *primary key* [22].



2.7 Pembelian

Pembelian adalah proses menukar uang yang sah sebagai alat pembayaran dengan barang yang dilakukan oleh beberapa pihak atau perusahaan. Dalam hal ini, pembeli menerima barang yang diinginkan dan penjual menerima keuntungan atas barang yang telah terjual. Dalam bisnis perdagangan, pembelian dilakukan untuk memperoleh barang dagangan atau stok barang yang nantinya akan dijual kembali kepada pelanggan [23].

Pembelian merupakan proses pengadaan barang atau jasa untuk mencapai tujuan organisasi atau perusahaan. Departemen pembelian memiliki tujuan utama untuk memastikan kualitas dan nilai produk perusahaan, meminimalkan pemakaian modal untuk pemasokan stok barang, menjaga aliran barang masuk dan keluar, dan memperkuat daya saing organisasi. Pembelian juga melibatkan penerimaan dan proses permintaan resmi, menawarkan dan mencari barang, evaluasi penawaran, pemeriksaan barang yang diterima, dan pengawasan penyimpanan dan pemakaian yang sesuai [24].

Fungsi utama dari pembelian adalah untuk menyediakan barang dan jasa yang dibutuhkan oleh perusahaan pada waktu, harga, dan kualitas yang tepat. Proses pembelian melibatkan serangkaian tahapan yang harus dilalui, seperti perkiraan kebutuhan dasar produksi oleh departemen produksi. Pembelian memegang peran penting dan memiliki dampak besar bagi setiap perusahaan, terutama jika perusahaan beroperasi pada skala besar. Dalam hal ini, pembelian

menjadi proses yang kompleks dan harus dikendalikan dengan baik karena terkait dengan kondisi keuangan perusahaan [25].

2.8 Penjualan

Penjualan adalah proses pertukaran barang atau jasa antara penjual atau perusahaan dengan konsumen melalui pembayaran dengan uang atau alat pembayaran lain yang sah. Dalam hal ini, penjual memberikan barang atau jasa kepada pembeli dan menerima pembayaran dalam bentuk uang atau alat pembayaran lain [23]. Kegiatan penjualan meliputi penjualan barang dan jasa, baik melalui pembayaran kredit maupun pembayaran tunai. Dalam transaksi penjualan kredit, setelah pesanan pelanggan terpenuhi melalui pengiriman barang atau penyediaan jasa, perusahaan memiliki piutang terhadap pelanggan selama jangka waktu yang disepakati. Namun, dalam sistem penjualan tunai, barang atau jasa hanya akan diserahkan setelah perusahaan menerima pembayaran tunai dari pembeli [26].

2.9 Persediaan

Barang-barang yang disimpan oleh suatu perusahaan untuk dijual sebagai bagian dari operasi bisnis biasa disebut sebagai persediaan. Kecuali untuk beberapa organisasi atau perusahaan jasa, persediaan merupakan aset yang diperlukan dan penting bagi perusahaan. Penting untuk melakukan pemeriksaan yang cermat pada persediaan karena hal ini merupakan bagian penting dari aset operasi dan mempengaruhi langsung penentuan laba [23]. Untuk suatu perusahaan, persediaan adalah faktor penting untuk memastikan kelancaran produksi. Jika terjadi kelebihan persediaan bahan baku, hal ini akan mengakibatkan biaya tambahan yang besar, dan jika disimpan terlalu lama, dapat menyebabkan kerusakan pada bahan baku. Sementara itu, jika jumlah persediaan terlalu sedikit, maka dapat mengganggu proses produksi dan merugikan perusahaan karena kehilangan kesempatan untuk memperoleh laba jika permintaan melebihi perkiraan. Oleh karena itu, bahan baku harus direncanakan dengan baik sebagai input yang esensial bagi suatu perusahaan [27].

2.10 Hutang

Hutang adalah sumber modal eksternal serta sifatnya sementara pada perusahaan dan harus dibayar kembali oleh perusahaan yang bersangkutan. Ini adalah kewajiban untuk mentransfer harta atau memberikan jasa di masa depan. Dengan kedua definisi tersebut, dapat disimpulkan bahwa

hutang adalah kewajiban finansial keuangan yang harus dibayar kembali dalam bentuk uang, barang, atau jasa pada saat jatuh tempo kepada pihak yang memberikan pinjaman [28].

Berdasarkan waktu atau jatuh tempo pembayaran, hutang terbagi menjadi dua jenis, yaitu Hutang Jangka Pendek dan Hutang Jangka Panjang [28]:

1. Hutang Jangka Pendek

Hutang jangka pendek adalah hutang yang jatuh tempo pembayarannya dalam satu tahun selama siklus akuntansi operasi normal perusahaan, dengan memanfaatkan aktiva lancar atau hasil pembentukan kewajiban lancar lain. Oleh karena itu, dapat dikatakan bahwa "kewajiban lancar" adalah hutang yang harus dibayar dalam satu tahun. Ini memiliki manfaat bagi perusahaan, seperti biaya yang lebih murah dan fleksibel. Hal ini disebut fleksibel karena dapat digunakan sesuai kebutuhan dalam jangka waktu pendek.

2. Hutang Jangka Panjang

Hutang Jangka Panjang adalah hutang dengan jatuh tempo pembayaran lebih dari satu tahun sejak tanggal neraca. Hutang jangka panjang memenuhi kebutuhan dana yang tidak hanya berasal dari aktiva lancar. Kewajiban ini biasanya terjadi karena kebutuhan dana yang diperlukan untuk membeli aktiva tetap tambahan, meningkatkan modal kerja tetap, membeli perusahaan lain atau untuk membayar hutang lain.

UNIVERSITAS
MIKROSKIL