

BAB II TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem didefinisikan sebagai sekumpulan prosedur yang saling terkait dan terhubung yang digunakan untuk melakukan tugas bersama. Dengan menggabungkan dukungan *software*, *hardware*, dan bantuan *brainware* melalui serangkaian prosedur yang akan membuat sistem yang berguna bagi pengguna [1].

Model umum sebuah sistem adalah *input*, proses, dan *output* dengan karakteristik sebagai berikut [2]:

1. Komponen Sistem

Terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan yang dapat berupa subsistem. Subsistem memiliki sifat dari sistem menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem

Lingkungan luar sistem yang bisa bersifat menguntungkan dan juga dapat merugikan sistem itu sendiri. Dengan demikian, lingkungan luar harus tetap dijaga dan dipelihara.

4. Penghubung Sistem

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung suatu sistem atau *interface*. Penghubung ini akan memungkinkan sumber-sumber daya yang mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain yang melalui penghubung tersebut.

5. Masukan Sistem

Energi yang dimasukkan ke sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan sinyal.

6. Keluaran Sistem

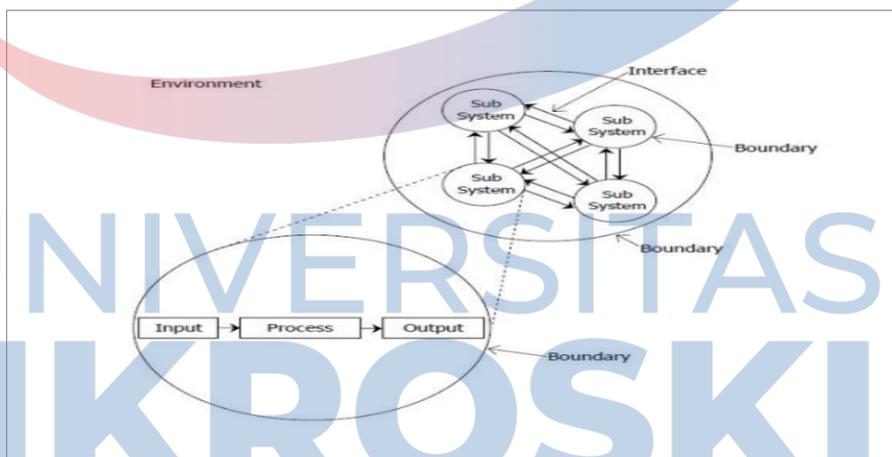
Hasil energi yang diolah dan diklasifikasikan akan menjadi keluaran yang sangat berguna. Keluaran yang dihasilkan adalah informasi untuk syarat pengambilan keputusan.

7. Pengolahan Sistem

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

8. Sasaran Sistem

Suatu sistem memiliki tujuan dan sasaran pasti dan bersifat deterministik. Kalau sistem tidak memiliki sasaran, maka operasi suatu sistem tidak ada gunanya. Suatu sistem dikatakan berhasil jika mencapai sasaran atau tujuan yang telah direncanakan.



Gambar 2.1 Karakteristik Sistem

2.1.2 Informasi

Informasi merupakan hasil pengolahan data dari satu atau lebih sumber kemudian diolah untuk memberikan nilai, makna, dan manfaat. Dalam proses pengolahan data, agar dapat menghasilkan informasi, proses verifikasi harus dilakukan secara akurat, spesifik, dan tepat waktu agar memberikan nilai dan pemahaman kepada pengguna [1].

Suatu informasi tidak dapat persis ditafsir keuntungannya dengan suatu nilai uang, tetapi dapat ditafsir nilai efektifitasnya yang didasarkan atas 10 (sepuluh) sifat, yaitu [2]:

1. Mudah Diperoleh

Sifat ini menunjukkan informasi dapat diperoleh dengan mudah dan cepat.

2. Luas dan Lengkap

Sifat ini menunjukkan lengkapnya isi informasi. Hal ini tidak berarti hanya mengenai volumenya, tetapi juga mengenai keluaran informasinya.

3. Ketelitian

Sifat ini menunjukkan minimnya kesalahan dalam informasi. Dalam hubungan dengan volume data yang besar biasanya terjadi dua jenis kesalahan, yakni kesalahan pencatatan dan kesalahan perhitungan.

4. Kecocokan

Sifat ini 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 ada keterlambatan jika ada yang sedang ingin mendapatkan informasi. Masukan, pengolahan, dan pelaporan keluaran kepada pemakai biasanya tepat waktu.

6. Kejelasan

Sifat ini menunjukkan keluaran informasi yang bebas dari istilah-istilah yang tidak jelas.

7. Keluwesan

Sifat ini berhubungan dengan penyesuaian keluaran informasi. Tidak hanya dengan beberapa keputusan, tetapi juga dengan beberapa pengambilan keputusan.

8. Dapat Dibuktikan

Sifat ini menunjukkan kemampuan beberapa pemakai informasi untuk menguji keluaran informasi dan sampai pada kesimpulan yang sama.

9. Tidak Ada Prasangka

Sifat ini berhubungan dengan tidak adanya keinginan untuk mengubah informasi guna mendapatkan kesimpulan yang telah dipertimbangkan.

10. Dapat Diukur

Sifat ini menunjukkan hakikat informasi yang dihasilkan dari sistem informasi formal.

2.1.3 Sistem Informasi

Sistem informasi merupakan sistem internal organisasi, yang menjumlahkan kebutuhan pemrosesan transaksi harian untuk mendukung fungsi manajemen operasi organisasi dan kegiatan strategis organisasi, sehingga dapat memberikan laporan yang diperlukan kepada pihak eksternal tertentu [2].

Terdapat definisi sistem informasi menurut beberapa ahli, yaitu [3] :

1. Menurut Steven Alter (1992)

Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan sebuah organisasi.

2. Menurut Bodnar dan Hopwood (1993)

Sistem informasi adalah kumpulan perangkat keras dan lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.

3. Menurut Gelinas, Oram, dan Wiggins (1990)

Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer yang manual dan dibuat untuk menghimpun, menyimpan, dan mengelola data, serta menyediakan informasi keluaran kepada para pemakai.

Berdasarkan ketiga definisi sistem informasi di atas, dapat disimpulkan bahwa sistem informasi merupakan kombinasi dari perangkat lunak (*software*), perangkat keras (*hardware*), dan sumber daya manusia yang dirancang ke sebuah sistem yang berguna untuk mengolah data menjadi informasi kepada pemakai untuk mencapai tujuan.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yaitu [2]:

1. Blok masukan (*input block*)

Input block mewakili data yang masuk ke dalam sistem informasi. Yang dimaksud dengan *input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

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

3. Blok keluaran (*output block*)

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

4. Blok teknologi (*technology block*)

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, serta membantu pengendalian sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan *Database Management System (DBMS)*.

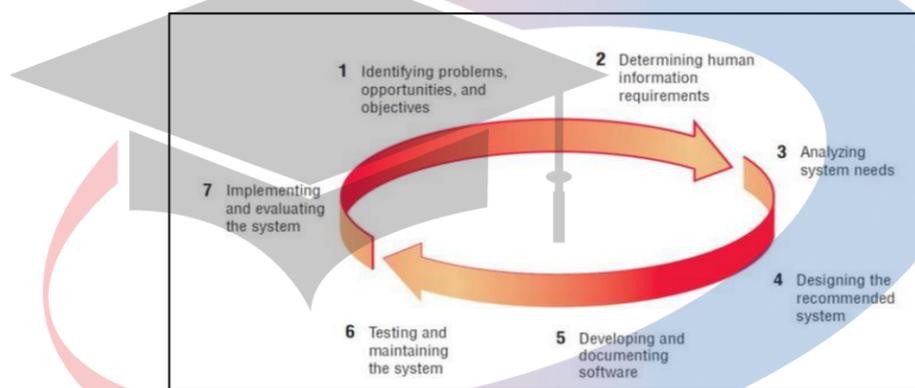
6. Blok kendali (*control block*)

Banyak hal dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan pada sistem itu sendiri, ketidak-efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal

yang dapat merusak sistem dicegah dan bila terlanjur terjadi, maka kesalahan-kesalahan dapat dengan cepat diatasi.

2.2 SDLC (System Development Life Cycle)

System Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang di mana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik [4].



Gambar 2.2 SDLC (*System Development Life Cycle*)

Terdapat tujuh tahap dalam siklus hidup pengembangan sistem (SDLC) yaitu sebagai berikut [4]:

1. Mengidentifikasi Masalah, Peluang, dan Tujuan

Pada tahap pertama dari siklus hidup pengembangan sistem, penganalisis mengidentifikasi masalah, peluang, dan tujuan yang hendak dicapai. Penganalisis akan melihat apa yang terjadi dalam bisnis. Kemudian, bersama anggota organisasi lainnya menentukan secara tepat masalah-masalah tersebut. Setelah itu, penganalisis harus dapat meyakinkan bahwa dengan memanfaatkan peluang penggunaan sistem informasi terkomputerisasi dapat memberikan peningkatan yang memungkinkan bisnis untuk mencapai sisi kompetitif sesuai tujuan - tujuan organisasi.

2. Menentukan syarat-syarat informasi

Pada tahap kedua dari siklus hidup pengembangan sistem, penganalisis memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Metode yang dapat dipakai untuk menentukan syarat-

syarat informasi yang tepat adalah dengan melibatkan interaksi langsung dengan pemakai. Beberapa hal yang dilakukan pada tahap ini adalah berupa menentukan sampel, memeriksa data mentah, wawancara, mengamati perilaku pembuat keputusan, mengamati lingkungan kantor, dan membuat *prototyping*. Orang-orang yang terlibat dalam tahap ini adalah penganalisis dan pemakai.

3. Menganalisis kebutuhan sistem

Pada tahap ketiga dari siklus hidup pengembangan sistem, penganalisis menggunakan beberapa teknik untuk menganalisis kebutuhan-kebutuhan sistem. Sekali lagi, perangkat dan teknik-teknik tertentu akan membantu penganalisis menentukan kebutuhan. Perangkat yang dimaksud ialah penggunaan diagram aliran data untuk menyusun daftar *input*, proses dan *output* fungsi bisnis dalam bentuk grafik terstruktur. Dari diagram aliran data, dikembangkan suatu kamus data berisikan daftar seluruh item data yang digunakan dalam sistem.

4. Merancang sistem yang direkomendasikan

Dalam tahap desain dari siklus hidup pengembangan sistem, penganalisa sistem menggunakan informasi-informasi yang terkumpul sebelumnya untuk mencapai desain sistem informasi yang logis. Penganalisis merancang prosedur data masukan sedemikian rupa sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Selain itu, penganalisis menggunakan teknik-teknik bentuk dan perancangan layar tertentu untuk menjamin keefektifan input sistem informasi.

5. Mengembangkan dan mendokumentasikan perangkat lunak

Pada tahap kelima dari siklus hidup pengembangan sistem, penganalisis bekerja sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang diperlukan. Beberapa teknik terstruktur untuk merancang dan mendokumentasikan perangkat lunak meliputi rencana terstruktur dan *pseudocode*. Selama tahap ini, penganalisis juga bekerja sama dengan pemakai untuk mengembangkan dokumentasi perangkat lunak yang efektif.

6. Menguji dan mempertahankan sistem

Pada tahap keenam dari siklus hidup pengembangan sistem, sebelum sistem informasi dapat digunakan, maka harus dilakukan pengujian terlebih dahulu.

Sebagian pengujian dilakukan oleh pemrogram dan oleh penganalisis. Mempertahankan sistem dan pendokumentasian dimulai pada tahap ini dan dilakukan secara rutin selama sistem informasi dijalankan.

7. Mengimplementasikan dan mengevaluasi sistem

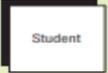
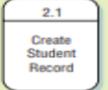
Pada tahap terakhir dari siklus hidup pengembangan sistem, penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan sistem. Selain itu, penganalisis perlu merencanakan konversi perlahan dari sistem lama ke sistem baru. Proses ini mencakup pengubahan *file-file* dari format lama ke sistem baru atau membangun suatu sistem basis data, memasang peralatan, dan membawa sistem baru untuk diproduksi. Evaluasi merupakan bagian akhir dari tahap terakhir ini melalui diskusi atau pembahasan yang menyatakan pemakai telah puas dengan sistem informasi yang dikembangkan.

2.3 Teknik Pengembangan Sistem

2.3.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafik dari sistem. DFD menggambarkan komponen-komponen sistem, aliran data antara komponen-komponen tersebut, asal, tujuan dan penyimpanan dari data tersebut. Dengan menggunakan *Data Flow Diagram* (DFD), maka pengguna sistem yang belum memahami bidang komputer dapat memahami sistem yang sedang berjalan [4].

Terdapat empat simbol dasar yang digunakan untuk memetakan gerakan DFD yaitu [4]:

Symbol	Meaning	Example
	Entity	
	Data Flow	
	Process	
	Data Store	

Gambar 2.3 Simbol Data Flow Diagram (DFD)

1. Entitas

Entitas merupakan kesatuan di luar lingkungan sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang berada di lingkungan luar, yang akan memberikan *input* atau menerima *output* dari sistem.

2. Arus Data

Arus data ditunjukkan dengan simbol suatu panah. Arus data ini mengalir di antara proses, simpanan data, dan entitas.

3. Proses

Proses adalah kegiatan arus kerja yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu aliran data yang masuk ke dalam proses untuk menghasilkan aliran data yang akan keluar dari proses. Kesalahan proses yang harus dihindari pada saat menggambarkan proses adalah menghindari *Black Hole* (proses mempunyai *input*, tetapi tidak menghasilkan *output*) dan *Miracle* (proses menghasilkan *output* tetapi tidak menerima *input*).

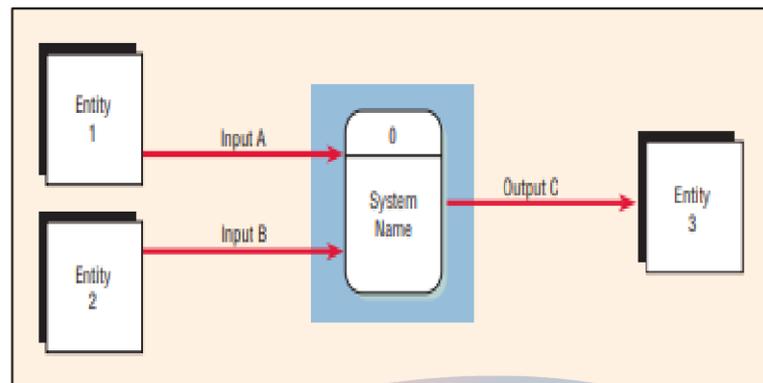
4. Simpanan Data

Simpanan data dapat berupa *file*, *database*, arsip, tabel acuan manual, agenda, dan buku.

Data Flow Diagram dapat dibagi menjadi tiga level, yaitu [4]:

1. Diagram Konteks

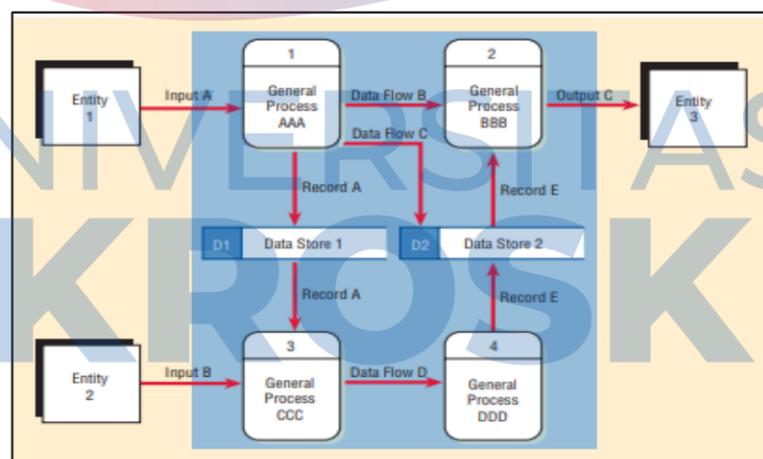
Diagram konteks adalah tingkatan tertinggi dalam data *flow* diagram dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal ditunjukkan pada diagram konteks berikut aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data menuju dan dari sistem diketahui penganalisis dari wawancara dengan pengguna dan sebagai hasil analisis dokumen.



Gambar 2.4 Diagram Konteks

2. Diagram 0 (DFD Level 0)

Diagram 0 adalah pengembangan dari diagram konteks dan bisa mencakup sampai sembilan proses. Memasukkan lebih banyak proses pada level ini akan terjadi dalam suatu diagram yang kacau yang sulit dipahami. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data utama dari sistem (mewakili *file-file master*) dan semua entitas eksternal dimasukkan ke dalam diagram 0.

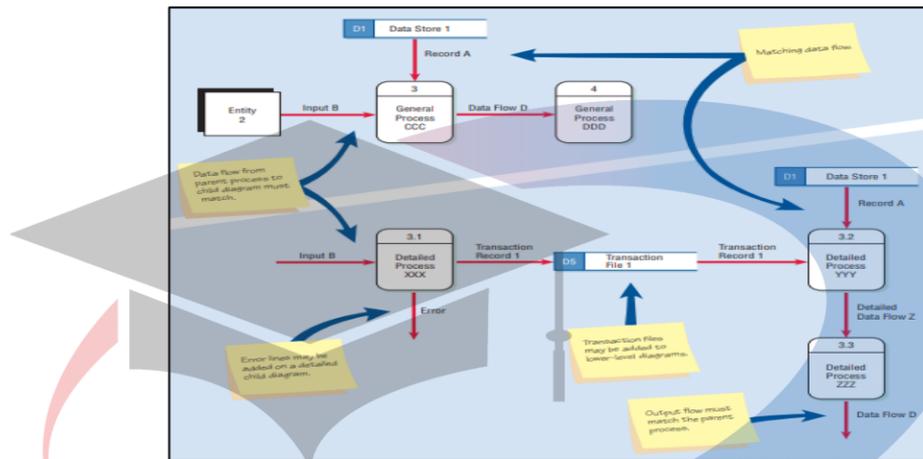


Gambar 2.5 Diagram Level 0

3. Diagram Rinci (DFD Level Anak)

Setiap proses dalam diagram 0 bisa dikembangkan untuk menciptakan diagram anak yang lebih mendetil. Proses pada diagram 0 yang dikembangkan itu disebut *parent process* (proses induk) dan diagram yang dihasilkan disebut *child diagram* (diagram anak). Aturan utama untuk menciptakan diagram anak, keseimbangan vertikal, menyatakan bahwa suatu diagram anak tidak bisa

menghasilkan keluaran atau menerima masukan di mana proses induknya juga tidak menghasilkan atau menerima. Semua aliran data yang menuju atau keluar dari proses induk harus ditunjukkan mengalir ke dalam atau keluar dari diagram anak. Diagram anak ditetapkan nomor yang sama seperti proses induknya dalam diagram level 0.



Gambar 2.6 Diagram Rinci

Beberapa kesalahan umum yang dibuat saat menggambar DFD adalah sebagai berikut [4]:

1. Lupa memasukkan suatu aliran data atau mengarahkan kepala anak panah pada arah yang salah.
2. Menghubungkan penyimpanan data dan entitas-entitas eksternal secara langsung satu sama lain.
3. Aliran data atau proses-proses pemberian label yang tidak tepat.
4. Memasukkan lebih dari sembilan proses pada diagram aliran data. Dan memiliki terlalu banyak proses yang menciptakan suatu diagram yang kacau akan memusingkan untuk dibaca dan malah menghalangi komunikasi.
5. Mengabaikan aliran data.
6. Menciptakan analisis yang tidak seimbang. Masing-masing diagram anak harus memiliki masukan dan aliran data keluaran yang sama seperti proses induk.

2.3.2 Fishbone Diagram

Diagram tulang ikan (*fishbone diagram*) atau *Ishikawa diagram* adalah alat yang membantu mengidentifikasi, mengklasifikasikan, dan menampilkan berbagai

kemungkinan penyebab masalah atau karakteristik kualitas tertentu. Diagram ini menggambarkan hubungan antara masalah dan semua penyebab yang mempengaruhi masalah. Diagram tulang ikan atau *fishbone* adalah salah satu cara untuk meningkatkan kualitas pemecahan masalah dan pengambilan keputusan. Disebut sebagai diagram tulang ikan karena diagram tersebut berbentuk mirip dengan tulang ikan yang moncong kepalanya menghadap ke kanan. Diagram ini akan menunjukkan dampak atau akibat dari masalah tersebut, dengan berbagai penyebabnya. Efek atau akibat digambarkan sebagai moncong kepala ikan, sedangkan tulangnya diisi oleh sebab-sebab sesuai dengan pendekatan permasalahannya [5].

Langkah-langkah dalam Menggunakan Metode Tulang Ikan, yaitu [5]:

1. Menyiapkan sesi tulang ikan (*fishbone*).
 Dalam hal ini menyiapkan siapa yang akan menjadi fasilitator/moderator kegiatan pemantauan, siapa yang menjadi notulen, dan berbagi fungsi lain yang diperlukan.
2. Mengidentifikasi akibat suatu masalah.
 Dalam rangka mengidentifikasi penyebab suatu keadaan yang tidak diharapkan adalah sebagai berikut:
 - a. Dimulai dengan pernyataan masalah-masalah penting dan mendesak untuk diselesaikan.
 - b. Menuliskan pernyataan masalah pada kepala ikan, yang merupakan akibat (*effect*). Pada sisi sebelah kanan dari kertas (kepala ikan), kemudian gambarkan tulang belakang dari kiri ke kanan dan tempatkan pernyataan masalah itu dalam kotak.
3. Mengidentifikasi berbagai kategori sebab utama.
 Menuliskan faktor penyebab utama (sebab-sebab) yang mempengaruhi masalah kualitas sebagai tulang besar yang diletakkan dalam kotak. Faktor-faktor penyebab atau kategori-kategori utama dapat dikembangkan melalui stratifikasi ke dalam pengelompokan dari faktor: manusia, mesin, peralatan, material, metode kerja, lingkungan kerja, pengukuran, dll.
4. Menemukan sebab-sebab potensial dengan cara sumbang saran (*brainstorming*).

Brainstorming digunakan untuk menguraikan sebab-sebab yang selalu ada pada setiap kategori. Dalam hal ini menentukan bersama sebab-sebab yang ditemukan harus ditempatkan dalam *fishbone* diagram, yaitu tentukan di bawah kategori yang mana gagasan tersebut harus ditempatkan.

5. Mengkaji dan menyepakati sebab-sebab yang paling mungkin.

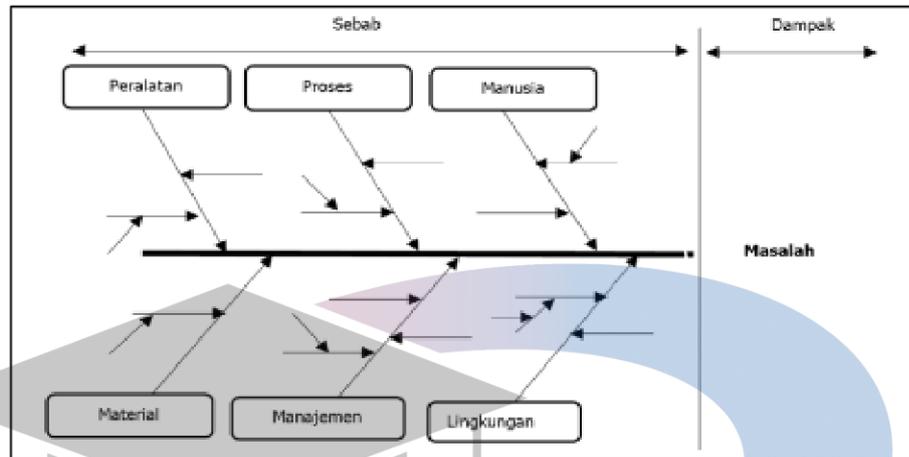
Setelah setiap kategori diisi, dengan mencari sebab yang paling mungkin di antara semua sebab-sebab dan sub-subnya. Jika ada sebab-sebab yang muncul pada lebih dari satu kategori, kemungkinan merupakan petunjuk sebab yang paling mungkin. Kaji kembali sebab-sebab yang telah didaftarkan (sebab yang paling memungkinkan) dan dipertanyakan. Berikut beberapa pendekatan untuk merumuskan faktor-faktor utama dalam mengawali pembuatan *fishbone* diagram [5]:

- a. Pendekatan *The 4 M's* (digunakan untuk industri manufaktur). Faktor-faktor utama yang bisa dijadikan acuan menurut pendekatan ini adalah 1) *Machine (Equipment)*, 2) *Method (Process/Inspection)*, 3) *Material (Raw, Consumables, dan lain-lain)*, dan 4) *Man power*.
- b. Pendekatan *The 8 P's* (digunakan pada industri jasa). Menurut pendekatan ini, ada setidaknya 8 hal yang bisa dijadikan acuan sebagai faktor utama, antara lain 1) *People*, 2) *Process*, 3) *Policies*, 4) *Procedures*, 5) *Price*, 6) *Promotion*, 7) *Place/Plant*, dan 8) *Product*.
- c. Pendekatan *The 5 S's* (digunakan pada industri jasa). Pendekatan ini memberikan acuan 5 faktor utama, antara lain 1) *Surroundings* (lingkungan), 2) *Supplier*, 3) *System* (sistem pelayanan konsumen), 4) *Skill*, dan 5) *Safety* (keselamatan).
- d. Pendekatan *4P* (pendekatan manajemen pemasaran). Pendekatan yang menggunakan perspektif manajemen pemasaran untuk memberikan faktor utama yang bisa dijadikan acuan, yakni 1) *Price*, 2) *Product*, 3) *Place*, dan 4) *Promotion*.

6. Mencapai kesepakatan atas sebab-sebab yang paling mungkin.

Tentukan item yang penting dari setiap faktor dan tandai faktor penting tertentu yang signifikan pada karakteristik kualitas. Untuk mengetahui faktor-faktor

penyebab dari suatu masalah yang sedang dikaji dapat mengembangkan pertanyaan.



Gambar 2.7 Contoh Diagram *Fishbone*

2.3.3 PIECES Framework

PIECES Framework adalah kerangka yang dipakai untuk mengklasifikasikan suatu *problem*, *opportunities*, dan *directives* yang terdapat pada bagian *scope definition* analisa dan perancangan sistem [6].

Metode *PIECES Framework* yaitu [6] :

1. Keandalan (*Performance*) adalah suatu kumpulan sistem dalam menyelesaikan tugas dengan cepat sehingga sasaran dapat segera tercapai. Indikator yang mempengaruhi:

- a. *Throughput*, dimana sistem ini dinilai dari banyaknya kerja yang dilakukan pada beberapa periode waktu.
- b. *Respon time*, yaitu delay rata-rata antara transaksi dan respon dari transaksi tersebut.
- c. *Audibilitas*, yaitu kecocokan dimana keselarasan terhadap standar dapat diperiksa.
- d. Kelaziman komunikasi, yaitu tingkat dimana *interface* sandar, protokol, dan *bandwidth* digunakan.
- e. Kelengkapan, yaitu derajat dimana implementasi penuh dari fungsi yang diharapkan tercapai.

- f. Konsistensi, yaitu penggunaan desain dan teknik dokumentasi yang seragam pada keseluruhan proyek pengembangan perangkat lunak.
 - g. Toleransi kesalahan, yaitu kerusakan yang terjadi pada saat program mengalami kesalahan.
2. Informasi (*Information*) adalah hal penting karena dengan informasi tersebut pihak manajemen dan *user* dapat melakukan langkah selanjutnya. Apabila kemampuan sistem informasi baik, maka *user* akan mendapatkan informasi yang akurat, tepat waktu dan relevan sesuai dengan yang diharapkan.
- a. *Accuracy* (akurat) dimana informasi yang dihasilkan memiliki ketepatan yang tinggi.
 - b. Relevansi informasi, dimana informasi yang dihasilkan sesuai dengan kebutuhan.
 - c. Penyajian informasi, dimana informasi yang disajikan dalam bentuk yang sesuai dan mudah diinterpretasikan.
 - d. Fleksibilitas data, dimana informasi mudah disesuaikan dengan kebutuhan
3. Nilai Ekonomis dari sistem (*Economics*) Pemanfaatan biaya yang digunakan dari pemanfaatan informasi. Peningkatan terhadap kebutuhan informasi yang ekonomis dapat mempengaruhi pengendalian biaya dan peningkatan manfaat terhadap sistem informasi
4. Pengamanan dan Pengendalian (*Control and Security*) Analisis ini digunakan untuk membandingkan sistem yang dianalisa berdasarkan pada segi integritas sistem, kemudahan akses, dan keamanan data:
- a. Integritas, tingkat dimana akses ke perangkat lunak atau data oleh orang yang tidak berhak dapat dikontrol.
 - b. Keamanan, yaitu mempunyai mekanisme yang mengontrol dan melindungi program.
5. Tingkat Keefisienan (*Efficiency*) Efisiensi berhubungan dengan bagaimana sumber tersebut dapat digunakan secara optimal. Operasi pada suatu perusahaan dikatakan efisien atau tidak biasanya didasarkan pada tugas dan tanggung jawab dalam melaksanakan kegiatan.

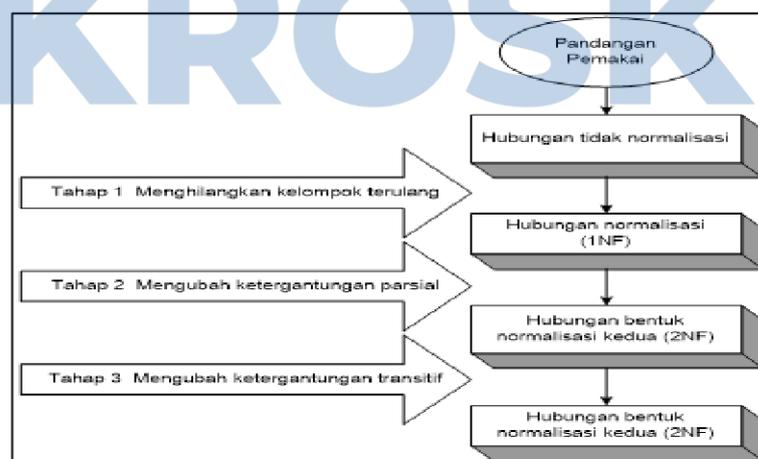
6. Pelayanan (*Service*) Peningkatan pelayanan memperlihatkan kategori yang beragam. Peningkatan pelayanan yang lebih baik bagi manajemen, *user* dan bagian lain merupakan simbol kualitas dari suatu sistem informasi.

2.3.4 Normalisasi

Normalisasi adalah transformasi tinjauan pemakai yang kompleks dan data tersimpan ke sekumpulan bagian-bagian struktur data yang kecil dan stabil. Di samping menjadi lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah diatur daripada struktur data lainnya [4].

Prosedur-prosedur yang dilakukan dalam tiga tahapan normalisasi adalah [4]:

1. Tahapan pertama dari proses meliputi menghilangkan semua kelompok yang terulang dan mengidentifikasi kunci utama. Untuk melakukan normalisasi tahap pertama ini, hubungan antar *field* perlu dipecah ke dalam dua atau lebih hubungan. Pada tahap ini, hubungan antar *field* tersebut ada yang mungkin sudah menjadi bentuk normalisasi ketiga.
2. Tahapan kedua dari proses normalisasi adalah menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain.
3. Tahapan ketiga dari proses normalisasi adalah mengubah ketergantungan transitif manapun. Suatu ketergantungan transitif adalah sesuatu di mana atribut bukan kunci tergantung pada atribut bukan kunci lainnya.

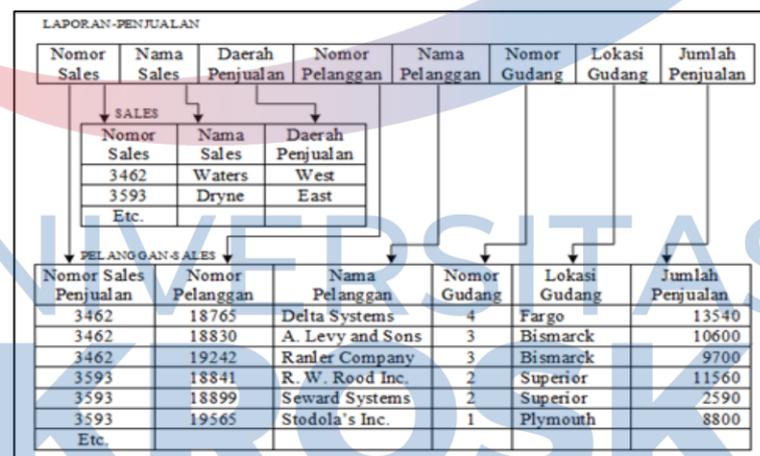


Gambar 2.8 Tahapan Normalisasi

Berikut ini adalah contoh normalisasi dari bentuk tidak normal hingga bentuk normal ketiga [4]:

1. Bentuk normalisasi pertama (1NF)

Langkah pertama dalam normalisasi hubungan adalah menghilangkan kelompok terulang. Gambar 2.9 menunjukkan hubungan tidak normal laporan penjualan dilakukan normalisasi dengan memisahkan hubungan ke dalam dua hubungan baru. Hubungan *sales* mengandung kunci utama nomor *sales* dan semua atribut yang tidak terulang (nama *sales* dan daerah penjualan). Hubungan kedua, pelanggan *sales* mengandung kunci utama dari hubungan *sales* (kunci utama dari *sales* adalah nomor *sales*) dan semua atribut yang merupakan bagian kelompok terulang (nomor pelanggan, nama pelanggan, nomor gudang, lokasi gudang dan jumlah penjualan). Dalam hubungan ini, menggunakan sebuah kunci gabungan yaitu nomor *sales* dan nomor pelanggan untuk mengakses informasi.



Gambar 2.9 Normalisasi 1NF

2. Bentuk normalisasi kedua (2NF)

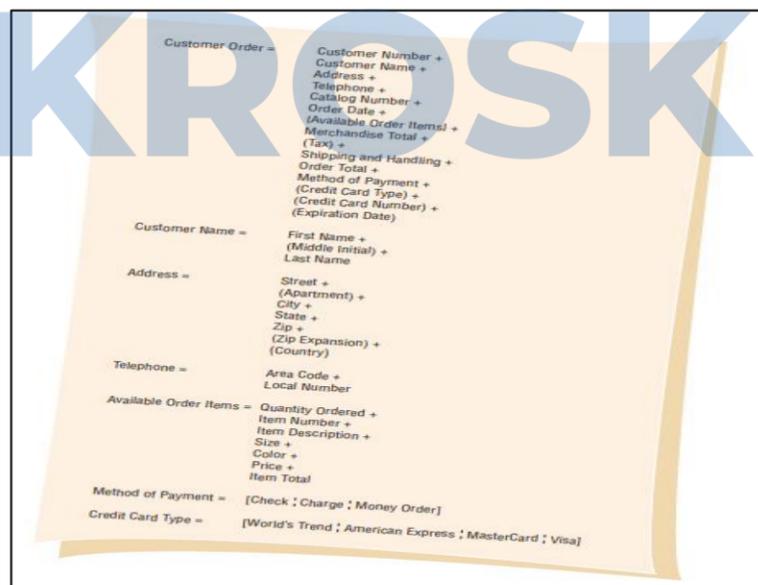
Dalam bentuk normalisasi kedua, semua atribut akan tergantung secara fungsional pada kunci utama. Oleh karena itu, langkah selanjutnya adalah menghilangkan semua atribut yang tergantung sebagian dan meletakkannya dalam hubungan lain. Gambar 2.10 menunjukkan hubungan pelanggan dan *sales* dipisah ke dalam dua hubungan baru yaitu penjualan dan gudang pelanggan.

Sebagai tambahan untuk dokumentasi serta mengurangi redundansi, kamus data bisa digunakan untuk [4]:

1. Menvalidasi *data flow diagram* dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan yang disimpan dalam *file-file*.
4. Mengembangkan logika untuk proses-proses *data flow diagram*.

Untuk menggambar struktur data dalam kamus data menggunakan notasi aljabar. Adapun simbol-simbol yang digunakan dalam notasi aljabar yaitu [4]:

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



Gambar 2.12 Kamus Data

2.3.6 Basis Data

Basis data merupakan kumpulan dari data yang saling berelasi satu dengan yang lainnya, yang digunakan untuk menyimpan informasi atau data, sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi masalah pada sistem yang memakai pendekatan berbasis berkas. Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System* (DBMS). DBMS memungkinkan para *user* untuk membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien dan orang yang memastikan bahwa basis data memenuhi tujuannya disebut administrator basis data [4].

Tujuan basis data yang efektif adalah [4]:

1. Memastikan bahwa data dapat dipakai diantara pemakai untuk berbagai aplikasi.
2. Memelihara data baik keakuratan maupun kekonsistenan.
3. Memastikan bahwa semua data yang diperlukan untuk aplikasi sekarang dan yang akan datang akan disediakan dengan cepat .
4. Membolehkan basis data untuk berkembang dan kebutuhan pemakai untuk berkembang.
5. Membolehkan pemakai untuk membangun pandangan personal-nya tentang data tanpa memperhatikan cara data disimpan secara fisik.

Tujuan yang telah disebutkan diatas mengingatkan tentang keuntungan dan kerugian pendekatan basis data. Pemakaian basis data berarti bahwa data perlu disimpan hanya sekali. Basis data juga membantu tercapainya integritas data, karena mengubah data yang diselesaikan lebih mudah dan dapat dipercaya jika data muncul hanya sekali dalam banyak *file* berbeda. Keuntungan pendekatan basis data yang memperbolehkan pemakai untuk memiliki pandangan sendiri mengenai data. Pemakai tidak perlu memperhatikan struktur sebenarnya pada basis data atau penyimpanan bentuk fisiknya. Sedangkan kerugian pendekatan basis data adalah bahwa semua data disimpan dalam satu tempat. Hal itu akan mengakibatkan data dapat terdampak bencana baik dari faktor *internal* atau *external* dan diperlukan adanya perencanaan *backup* yang baik [4].

2.4 Website

Website adalah suatu kumpulan *hyperlink* yang menuju alamat ke satu alamat lainnya dengan bahasa *Hypertext Markup Language* (HTML). Sebuah web atau situs dapat diartikan sebagai kumpulan halaman yang digunakan untuk menampilkan teks, gambar diam atau bergerak, animasi, suara, dan kombinasi semuanya., baik yang bersifat statis maupun dinamis dengan membentuk rangkaian bangunan yang saling berhubungan [7].

Jenis-jenis *website* berdasarkan sifatnya ada dua macam, yaitu [7]:

1. *Website* Dinamis, adalah *website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain *PHP*, *ASP*, *NET*, dan memanfaatkan *database MySQL* atau *Microsoft SQL Server*.
2. *Website* Statis, adalah *website* yang kontennya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah *HTML* dan belum memanfaatkan *database*.

2.5 Teknologi Website Yang Digunakan

2.5.1 HTML

Menurut Henderson (2009:232), *HTML* (*Hyper Text Mark Up Language*) merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman web. *HTML* berfungsi untuk mempublikasi dokumen online. Statement dasar dari *HTML* disebut *tags*. Sebuah *tag* dinyatakan dalam sebuah kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen haruslah dibuat berupa pasangan. Terdiri dari *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag* [8].

2.5.2 CSS

Menurut Henderson (2009:72), *CSS* (*Cascading Style Sheet*) adalah bahasa-bahasa yang merepresentasikan halaman web. Seperti warna, *layout*, dan *font*. Dengan menggunakan *CSS*, seorang web developer dapat membuat halaman *web* yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan *CSS* biasanya terpisah dengan halaman *HTML*. Meskipun *CSS* dapat disisipkan di dalam

halaman *HTML*. Hal ini ditujukan untuk memudahkan pengaturan halaman *HTML* yang memiliki rancangan yang sama [8].

2.5.3 Bootstrap

Spurlock (2013:1), menyatakan bahwa *Bootstrap* adalah sebuah *framework* untuk *CSS* dan berupa produk *open source* yang dibuat oleh Mark Otto dan Jacob Thornton. Pada awalnya *Bootstrap* ini dibuat untuk membuat standarisasi *front end* untuk semua *programmer* di perusahaannya. *Bootstrap* telah berubah dari yang sebelumnya adalah *CSS-Driven proyek* ke sebuah *host* dari *JavaScript plugins* dan ikon yang dapat dengan mudah digunakan untuk formulir dan tombol [8].

2.5.4 JavaScript

Menurut Henderson (2009:256), *JavaScript* adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada web browser seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini* dan sebagainya. *JavaScript* pertama kali dikembangkan pada pertengahan dekade 90'an. Meskipun memiliki nama yang hampir serupa, *JavaScript* berbeda dengan bahasa pemrograman *Java*. Untuk penulisannya, *JavaScript* dapat disisipkan di dalam dokumen *HTML* ataupun dijadikan dokumen tersendiri yang kemudian diasosiasikan dengan dokumen lain yang dituju. *JavaScript* mengimplementasikan fitur yang dirancang untuk mengendalikan bagaimana sebuah halaman web berinteraksi dengan penggunanya [8].

2.5.5 MySQL

Menurut Kurniawan (2010:16), *MySQL* adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis. *MySQL* termasuk jenis *RDBMS (Relational Database Management Sistem)*. *MySQL* ini mendukung bahasa pemrograman *PHP*. *MySQL* juga mempunyai *Query* atau bahasa *SQL (Structured Query Language)* yang simple dan menggunakan *escape character* yang sama dengan *PHP* [8].

2.5.6 ReactJs

ReactJs merupakan salah satu pustaka (*library*) *JavaScript* yang dikembangkan dan juga dirilis oleh *Facebook* pada tahun 2013, dan menjadi pendukung *Facebook* [9].

React menggunakan *DOM* virtual yang membantu mengintegrasikan dengan aplikasi apa pun. *React* juga mengandalkan *JSX* dalam menyusun komponen dan membantu mengembangkan halaman *web* yang lebih ramah *SEO* dibandingkan dengan perpustakaan atau kerangka kerja *JavaScript* lainnya [9].

Kelebihan dari *ReactJs* yaitu [9] :

1. *DOM Binding*

React dapat mengikat pengikatan di beberapa area kode untuk menghubungkan elemen *DOM* ke fungsi.

2. Perpustakaan (*library*), bukan kerangka kerja (*framework*)

Bereaksi adalah perpustakaan dan bukan Kerangka kerja *JavaScript*. Ia menyediakan metode pengantar untuk mendefinisikan komponen *UI* dan dapat dikombinasikan dengan perpustakaan lain seperti *React-Redux* dengan mudah.

3. Data dan presentasi

Menyediakan pemisahan lengkap antara lapisan data dan presentasi, dan juga digunakan untuk penyimpanan berlangsung dalam waktu yang sangat singkat.

4. *Unidirection Data Flow*

yaitu menggunakan data yang mengikat, yang membuat kode lebih stabil dan yang perlu dilakukan pengembang untuk mengubah suatu objek adalah mengubah posisi dan menerapkan pembaruan.

5. Komponen yang dapat digunakan kembali

Meningkatkan efisiensi desain dan memudahkan *developer* untuk menangani *upgrade*. dan tidak akan mempengaruhi komponen lain, sehingga dapat menghemat banyak waktu selama pengembangan.

2.5.7 NodeJs

NodeJs adalah sebuah *runtime environment opensource* dan *cross platform* yang biasa digunakan untuk mengembangkan aplikasi *web*, walaupun *NodeJs* bukan merupakan *framework JavaScript* namun banyak modul dasarnya yang ditulis

menggunakan *JavaScript Runtime environment* yang dimiliki oleh *NodeJs* menerjemahkan *JavaScript* menggunakan *Google V8 JavaScript engine*. *NodeJs* memiliki ciri khas yang sangat unik yaitu kemudahan dimana *web server* dan kode program nya tidak terpisah seperti pada bahasa pemrograman *web* lainnya. Pada *NodeJs* pengembang dapat mengaktifkan *web server* memodifikasinya dan menampilkan *content* dalam satu *file* tanpa harus mengubah banyak konfigurasi [9].

2.6 Penjualan

Penjualan adalah kegiatan memindahkan barang yang dijual kepada *customer*. Kegiatan dari penjualan dimulai dari adanya pesanan dari *customer*, penerimaan barang oleh *customer*, pembuatan faktur penjualan, dan pencatatan laporan penjualan. Penjualan tunai merupakan salah satu bentuk transaksi dari barang dan jasa dalam transaksi penjualan secara tunai ini. Penjual barang menyerahkan barang langsung kepada pihak pembeli setelah pembeli memberikan uang kepada penjual. Sistem penjualan tunai lebih mudah pelaksanaannya dan prosesnya juga lebih cepat [10].

2.7 Pembelian

Pembelian adalah suatu usaha yang digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan. Secara umum definisi pembelian adalah usaha pengadaan barang atau jasa dengan tujuan yang akan digunakan sendiri, untuk kepentingan proses produksi maupun untuk dijual kembali, baik dengan atau tanpa proses, dalam proses pembelian yang ada, agar kegiatan pembelian dapat dilakukan dengan benar [11].

Fungsi pembelian bertanggung jawab untuk memperoleh informasi mengenai harga barang, menentukan pemasok yang dipilih dalam pengadaan barang, dan mengeluarkan pesanan pembelian kepada pemasok yang dipilih [11].

Proses pembelian adalah sebuah struktur interaksi antara orang-orang, peralatan, metode-metode, dan pengendalian yang dirancang untuk mencapai fungsi-fungsi utama yaitu [11]:

1. Menangani rutinitas pekerjaan yang berulang-ulang dari bagian pembelian dan departemen penerimaan.

2. Mendukung kebutuhan pengambilan keputusan dari orang-orang yang mengatur bagian pembelian dan penerimaan.
3. Membantu dalam penyiapan laporan internal dan eksternal.

2.8 Persediaan

Persediaan adalah bahan atau barang yang disimpan yang akan digunakan untuk memenuhi tujuan tertentu, misalnya untuk digunakan dalam proses produksi atau perakitan, untuk dijual kembali, atau untuk suku cadang dari suatu peralatan atau mesin. Persediaan dapat berupa bahan mentah, bahan pembantu, barang dalam proses, barang jadi, ataupun suku cadang. Dalam perusahaan dagang, persediaan hanya terdiri atas satu golongan, yaitu persediaan barang dagangan, yang merupakan barang yang dibeli untuk tujuan dijual kembali [11].

Persediaan sangat penting karena tanpa persediaan para pengusaha akan dihadapkan pada risiko bahwa perusahaannya pada suatu waktu tidak dapat memenuhi keinginan pelanggan yang memerlukan barang atau jasa yang dihasilkan. Persediaan dikatakan sangat penting bagi perusahaan, dikarenakan persediaan berguna untuk [12]:

1. Menghilangkan risiko keterlambatan datangnya barang.
2. Menghilangkan risiko dari produk yang dipesan tidak bagus atau rusak.
3. Mempertahankan stabilitas operasi perusahaan atau menjamin kelancaran arus produksi.
4. Memberikan pelayanan kepada pelanggan sebaik-baiknya, di mana keinginan pelanggan pada setiap waktu dapat terpenuhi atau memberi jaminan tetap tersedianya barang tersebut.

2.9 Hutang (*Liabilities*)

Hutang adalah kewajiban perusahaan kepada kreditur (*supplier*, bankir) dan pihak lainnya (karyawan, pemerintah). Kreditur dan pihak lainnya di sini memiliki hak atau klaim atas aktiva perusahaan [13].

Klasifikasi hutang dapat dikelompokkan menjadi beberapa yaitu [13]:

1. Hutang Usaha (*Accounts Payable*)

Hutang yang timbul pada saat barang atau jasa diterima sebelum melakukan pembayaran. Dalam transaksi perusahaan dagang, seringkali perusahaan membeli barang secara kredit dari pemasok untuk dijual kembali kepada para pelanggannya.

2. Hutang Bunga (*Interest Payable*)

Jumlah bunga yang terhutang kepada kreditur atas dana yang dipinjam. Dalam hal ini, debitur telah menikmati dana kreditur selama periode berjalan namun baru akan dibayarkan di periode akuntansi berikutnya sesuai dengan tanggal jatuh tempo pinjaman.

3. Hutang Pajak Penghasilan Karyawan (*Employees Income Taxes Payable*)

Jumlah pajak yang terhutang kepada pemerintah atas besarnya gaji karyawan yang terkena pajak penghasilan.

4. Hutang Pajak Penjualan (*Sales Taxes Payable*)

Hutang atas pajak yang dipungut dari pembeli ketika penjualan terjadi. Seperti kita ketahui bahwa sebagai besar produk yang kita beli dari toko pengecer dikenakan pajak penjualan.

5. Hutang Upah (*Wages Payable*)

Jumlah upah yang terhutang kepada karyawan atas manfaat yang telah diterima perusahaan melalui pemakaian jasa karyawan selama periode berjalan.

6. Hutang Wesel (*Notes Payable*)

Kewajiban dalam bentuk janji tertulis (wesel bayar) untuk membayar sejumlah uang tertentu (pokok berikut bunganya) pada waktu yang ditetapkan.