

BAB II TINJAUAN PUSTAKA

2.1. Jurnalistik

Secara harfiah, jurnalistik (*journalistic*) artinya kewartawanan atau hal-hal pemberitaan. Kata dasar jurnalistik adalah “jurnal” (*journal*), artinya laporan atau catatan, atau “jour” (bahasa Prancis) yang berarti “hari” (*day*) atau “catatan harian” (*diary*). Dalam bahasa Belanda, *journalistiek* artinya penyiaran catatan harian. Demikian dikemukakan sejumlah literatur tentang asal-usul kata dan pengertian jurnalistik. Secara konseptual, jurnalistik dapat dipahami dari tiga sudut pandang: sebagai proses, teknik, dan ilmu. Sebagai proses, jurnalistik adalah “aktivitas” mencari, mengolah, menulis, dan menyebarluaskan informasi kepada publik melalui media massa. Aktivitas ini dilakukan oleh wartawan (jurnalis). Sebagai teknik, jurnalistik adalah “keahlian” (*expertise*) atau “keterampilan” (*skill*) menulis karya jurnalistik (berita, artikel, feature) termasuk keahlian dalam pengumpulan bahan penulisan seperti peliputan peristiwa (*reportase*) dan wawancara. Sebagai ilmu, jurnalistik adalah “bidang kajian” mengenai pembuatan dan penyebaran informasi (peristiwa, opini, pemikiran, ide) melalui media massa. Jurnalistik termasuk ilmu terapan (*applied science*) yang dinamis dan terus berkembang sesuai dengan perkembangan teknologi informasi dan komunikasi dan dinamika masyarakat itu sendiri. Sebagai ilmu, jurnalistik termasuk dalam bidang kajian ilmu komunikasi, yakni ilmu yang mengkaji proses penyampaian pesan, gagasan, pemikiran, atau informasi kepada orang lain dengan maksud memberitahu, mempengaruhi, atau memberikan kejelasan. Jurnalistik adalah bagian dari kehidupan kita sehari-hari. Jika setiap hari kita membaca, mendengar, atau menonton program berita, maka sadar atau tidak sadar kita terlibat dalam dunia jurnalistik, minimal sebagai objek atau sasaran (*target audience*) dari para jurnalis. Jurnalistik: Keahlian Utama Wartawan Bagi wartawan atau jurnalis, memahami ilmu dan teknik jurnalistik tentu merupakan hal yang mutlak. Namun demikian, masyarakat pembaca, pendengar, atau pemirsa pun penting mengenal dan memahami jurnalistik, setidaknya dasar-dasarnya,

sehingga tidak menjadi objek pasif media massa, bahkan bisa menjadi pembaca, pendengar, dan penonton kritis dan aktif terhadap sajian berita yang disebarkan media. Secara praktis, jurnalistik adalah disiplin ilmu dan teknik pengumpulan, penulisan, dan pelaporan berita, termasuk proses penyuntingan dan penyajiannya. [1]

2.2. Berita

Berita dapat disebut sebagai menu utama media. Berita (*news*) merupakan laporan peristiwa/kejadian yang dapat berupa tulisan (teks), karya foto, suara, dan video. Berita dalam bentuk tulisan (teks) dan gambar (foto, ilustrasi, kartun) umumnya muncul di media tercetak (koran, majalah). Berita dalam bentuk suara (tanpa teks dan gambar/foto) ada di radio, berupa suara. Ya, suara saja. Adapun berita dalam bentuk video dan suara muncul di televisi. Namun kerap kita temui, berita yang ditampilkan hanya gambar dan tanpa suara reporter/presenter. Kini, perkembangan teknologi maju pesat yang ditandai hadirnya internet. Dalam internet, berita yang muncul meliputi tulisan (teks), karya foto, suara, video. Penyebutan jenis berita (*type of news*) yang berupa tulisan (teks), karya foto, suara, dan video tersebut didasarkan pada format (bentuk) secara umum. Adapun jenis berita yang dilihat dari sisi isi (topik, konten, bahasan) kita menemukan berita politik, ekonomi, olahraga, cuaca, wanita, fashion, dan sebagainya. Dari sisi geografis pun, penamaan jenis berita ini bertambah. Misalnya berita lokal (kota/kabupaten), berita regional (wilayah lebih luas, misal provinsi, kawasan pulau), berita nasional (mencakup negara), dan berita internasional (antarnegara). Jika melihat isi, kita dapat menemukan banyak lagi. Dari berita olahraga, kita dapat kembangkan berita sepakbola, tinju, atletik, renang, dan seterusnya. Tetapi, dalam perspektif jurnalistik, jenis-jenis berita meliputi *straight news*, *hard news*, *soft news*, *depth news*, *investigation news*, *interpretative news*, dan *opinion news*.

1. *Straight News* (Berita Langsung). Berita ini ditulis singkat dan lugas. Bahkan, apa adanya. Sebagian besar halaman depan surat kabar yang terbit

harian berisi berita jenis ini. Pun berita radio, televisi, dan media online sebagian berupa besar berita jenis ini.

2. *Hard News*. Jenis berita ini hampir mirip dengan dengan *straight news*, yaitu berita yang memiliki nilai lebih dari sisi aktualitas. Bisa jadi, amat penting untuk segera diketahui pembaca. Oleh karenanya, ada yang mengartikan *hard news* dengan berita serius dan menegangkan, misalnya kerusuhan, bom teroris, bencana, dan sejenisnya. Jadi, *hard news* lebih ke konten isinya.
3. *Soft News* (Berita Ringan). Sesuai namanya, berita jenis ini mempunyai nilai berita di bawah *hard news* dan tak sepenting *straight news*. Beritanya tidak menegangkan, tetapi tetap penting diketahui pembaca, misalnya ada wanita melahirkan anak kembar tiga di usia 60 tahun.
4. *Depth News* (Berita Mendalam). Sesuai namanya, berita ini tidak menampilkan peristiwa sekilas, namun lebih panjang dan lebih detail. Ada latar belakang, dampak, prosedur yang ditempuh, pandangan ahli, dan seterusnya. *Depth news* ini umum disajikan dalam tulisan feature.
5. *Investigative News* (Berita Investigasi). Berita yang disajikan melalui proses dandiolah berdasarkan penyelidikan. Data-data yang didapat melalui kerja layaknya "intelijen". Butuh waktu, keberanian, dan kadang-kadang melalui penyamaran. Misalnya pembobolan bank yang melibatkan banyak pihak (pegawai bank, pejabat pemerintah, pengawas keuangan, dan pihak terkait lainnya). Dalam kaitan ini, wartawan bertindak sebagai "detektif".
6. *Interpretative News* (Berita Interpretasi). Berita yang disusun berdasarkan penafsiran atau pendapat narasumber tentang suatu peristiwa, kejadian, atau masalah. Ada satu peristiwa, lalu wartawan mengembangkannya dengan mewawancarai pengamat atau narasumber yang punya keahlian dan kompetensi.
7. *Opinion News* (Berita Opini). Berita yang isinya menyajikan pendapat pakar, ahli, atau pengamat tentang suatu peristiwa atau masalah. Bahkan, isi pemikiran seorang pakar di sebuah acara seminar misalnya, dapat dijadikan berita.

Tujuh jenis berita yang diungkap di atas semuanya menyandarkan diri pada fakta. Dalam dunia jurnalistik harus mengandung unsur-unsur 5W+1H. [2]

2.3. Normalisasi

Normalisasi adalah transformasi tinjauan pemakaian yang kompleks dan data tersimpan ke sekumpulan bagian-bagian struktur data yang kecil dan stabil. Disamping itu menjadi lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah diatur dari pada struktur data lainnya.

Dalam normalisasi terdapat tiga tahapan yang digunakan, yaitu:

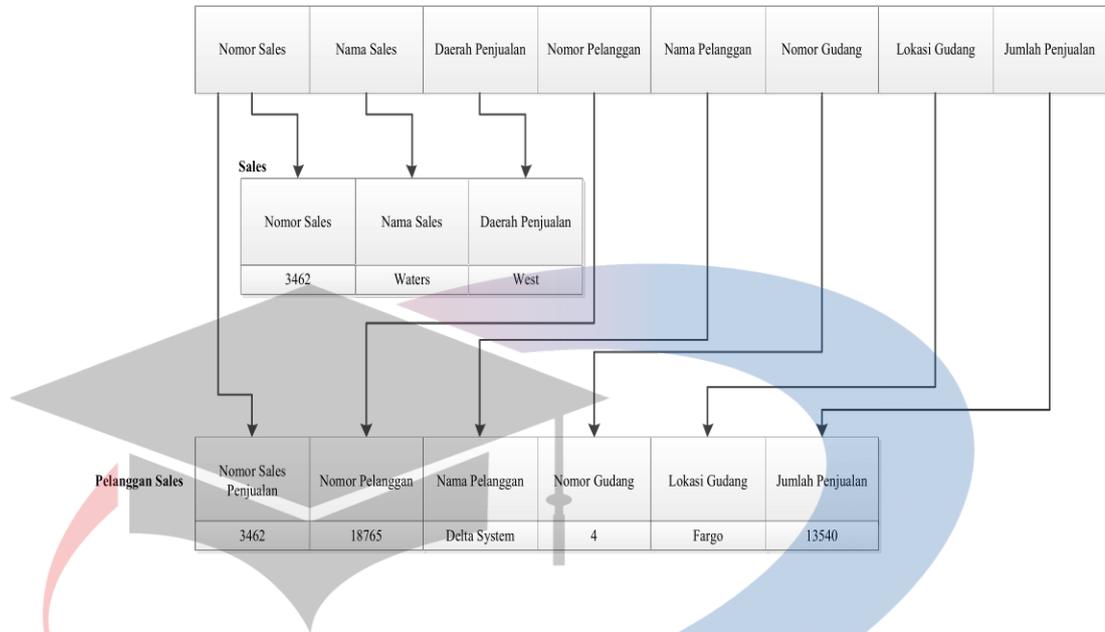
1. Tahapan pertama dari proses meliputi menghilangkan semua kelompok terulang dan mengidentifikasi kunci utama. Untuk mengerjakannya, hubungan perlu dipecah ke dalam dua atau lebih hubungan. Pada titik ini hubungan mungkin sudah menjadi bentuk normalisasi ketiga, bahkan lebih banyak tahap dan akan diperlukan untuk mentransformasi hubungan ke bentuk normalisasi ketiga.
2. Tahap kedua, menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain.
3. Tahap ketiga, mengubah ketergantungan transitif manapun. Suatu ketergantungan transitif adalah sesuatu dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya.

Misalkan normalisasi dilakukan terhadap Laporan Penjualan sebuah perusahaan dimana Laporan Penjualan tersebut memiliki atribut-atribut seperti Nomor Sales, Nama Sales, Daerah Penjualan, Nomor Pelanggan, Nama Pelanggan, Nomor Gudang, Lokasi Gudang, dan Jumlah Penjualan. Laporan penjualan adalah suatu hubungan tidak normal (*unnormalized relation*), karena memiliki kelompok berulang.

a. Bentuk Normalisasi Pertama (1NF)

Langkah pertama dalam normalisasi hubungan adalah menghilangkan kelompok terulang. Pada contoh diatas, hubungan tidak normal Laporan Penjualan akan

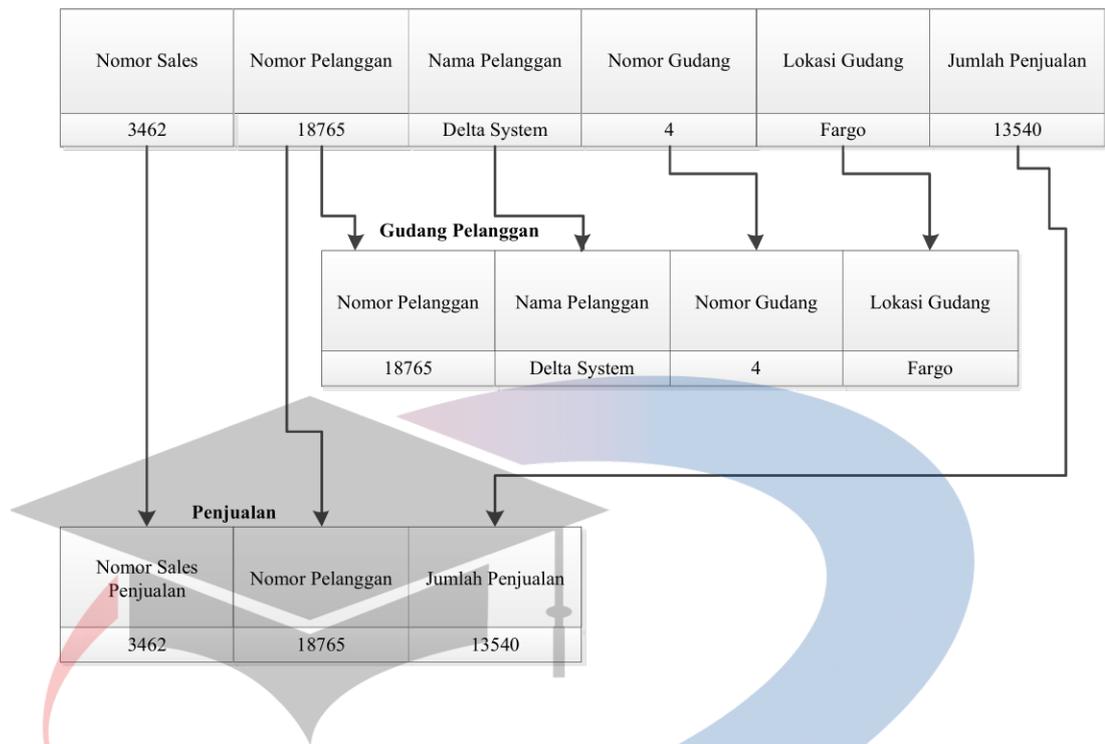
dipecah kedalam dua hubungan terpisah. Hubungan baru tersebut dinamakan Sales dan Pelanggan Sales.



Gambar 2. 1 Bentuk Normalisasi Pertama (1NF)

b. Bentuk Normalisasi Kedua (2NF)

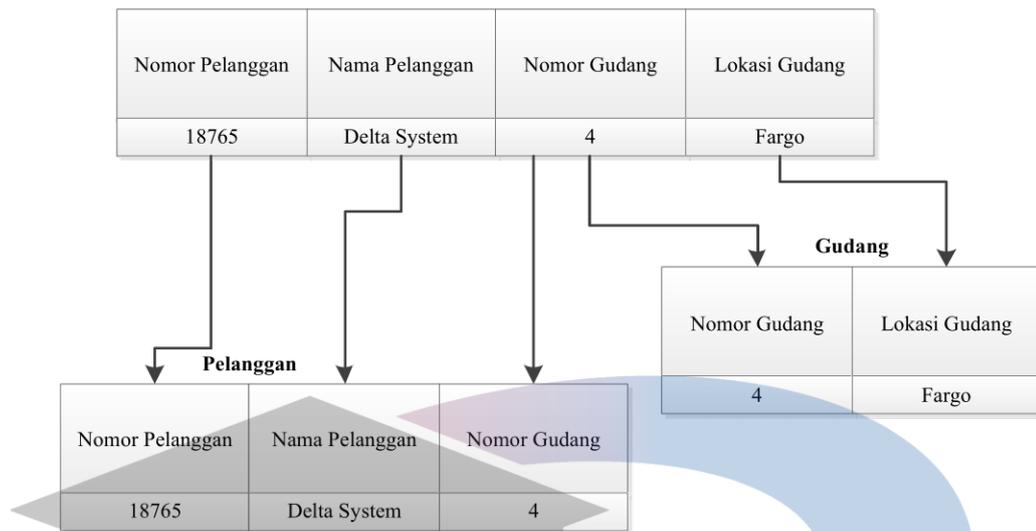
Dalam bentuk normalisasi kedua, semua atribut akan tergantung secara fungsional pada kunci utama. Oleh karena itu, langkah selanjutnya adalah menghilangkan semua atribut yang tergantung sebagian dan meletakkannya dalam hubungan lain. Pada contoh diatas, hubungan Pelanggan-Sales merupakan hubungan normalisasi pertama, tetapi tidak dalam bentuk ideal. Permasalahan muncul karena beberapa atribut tidak tergantung secara fungsional pada kunci utama (yaitu, nomor-sales, nomor-pelanggan) sehingga perlu dinormalisasikan kembali. Hubungan Pelanggan-Sales dipisahkan kedalam dua hubungan baru yaitu Penjualan dan Gudang-Pelanggan.



Gambar 2. 2 Bentuk Normalisasi Kedua (2NF)

c. Bentuk Normalisasi Ketiga (3NF)

Suatu hubungan normalisasi adalah bentuk normalisasi ketiga jika semua atribut kunci sepenuhnya tergantung secara fungsional pada kunci utama dan tidak terdapat ketergantungan transitif (bukan kunci). Pada contoh diatas menunjukkan ketergantungan dalam hubungan gudang-pelanggan. Untuk hubungan dalam bentuk normalisasi kedua, semua atribut harus tergantung pada kunci utama nomor-pelanggan , tetapi lokasi-gudang, juga tergantung secara nyata pada nomor-gudang. Untuk menyederhanakan hubungan ini maka perlu dilakukan normalisasi ketiga dimana hubungan Gudang-Pelanggan dipisah kedalam dua hubungan yaitu Pelanggan dan Gudang. [3]



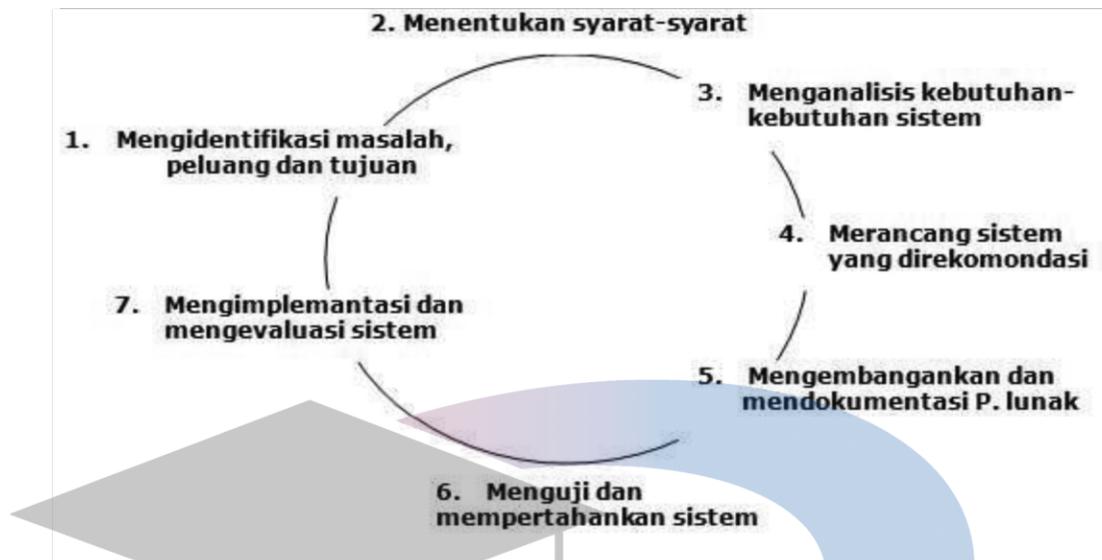
Gambar 2. 3 Bentuk Normalisasi Ketiga (3NF)

2.4. SDLC (*System Development Life Cycle*)

Sistem Development Life Cycle (SDLC) adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem.

Siklus hidup pengembangan sistem adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik.

Siklus hidup pengembangan sistem dibagi dalam tujuh tahap, seperti yang ditunjukkan dalam Gambar 2.4. Meskipun masing-masing tahap ditampilkan secara terpisah, namun tidak pernah tercapai sebagai satu langkah terpisah. Melainkan beberapa aktivitas muncul secara simultan, dan aktivitas tersebut dilakukan secara berulang-ulang. Lebih berguna lagi memikirkan bahwa SHPS bisa dicapai dalam tahap-tahap (dengan aktivitas berulang yang saling tumpang tindih satu sama lainnya dan menuju ke tujuan terakhir) dan tidak dalam langkah-langkah terpisah.



Gambar 2. 4 SDLC (System Development Life Cycle)

1. Mengidentifikasi masalah, peluang, dan tujuan.

Di tahap pertama dari siklus hidup pengembangan sistem ini, penganalisis mengidentifikasi masalah, peluang, dan tujuan-tujuan yang hendak dicapai. Tahap ini sangat penting bagi keberhasilan proyek, karena tidak seorangpun yang ingin membuang-buang waktu kalau tujuan masalah yang keliru. Tahap pertama ini berarti bahwa penganalisis melihat dengan jujur pada apa yang terjadi di dalam bisnis. Kemudian, bersama-sama dengan anggota organisasional lain, penganalisis menentukan dengan tepat masalah-masalah tersebut. Peluang adalah situasi dimana penganalisis yakin bahwa peningkatan bisa dilakukan melalui penggunaan sistem informasi terkomputerisasi. Peluang dan tujuan didapatkan dengan mengidentifikasi masalah-masalah yang terjadi di dalam perusahaan tersebut.

2. Menentukan syarat-syarat.

Tahap ini, penganalisis memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Diantara perangkat-perangkat yang digunakan untuk menetapkan syarat-syarat informasi di dalam bisnis diantaranya ialah menentukan sampel dan memeriksa data mentah, wawancara, mengamati perilaku pembuat keputusan dan lingkungan kantor dan *prototyping*.

3. Menganalisis kebutuhan-kebutuhan sistem.

Tahap berikut ini membantu penganalisis dalam menganalisis kebutuhan dari sistem dengan perangkat dan teknik-teknik tertentu. Perangkat yang dimaksud adalah penggunaan diagram alir data untuk menyusun daftar *input*, proses, dan output fungsi bisnis dalam bentuk grafik terstruktur. Dari diagram alir 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 logik. Penganalisis merancang prosedur data-entry sedemikian rupa sehingga data yang dimasukkan kedalam 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.

Dalam tahap kelima dari siklus hidup pengembangan sistem, penganalisis bekerja bersama-sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang diperlukan. Selain itu penganalisis juga bekerja sama dengan pemakai untuk mengembangkan dokumentasi perangkat lunak yang efektif.

6. Menguji dan mempertahankan sistem.

Sebelum sistem informasi dapat dipergunakan, maka harus dilakukan pengujian terlebih dahulu. Akan bisa menghemat biaya bila dapat menangkap adanya masalah sebelum sistem tersebut ditetapkan. Sebagian pengujian dilakukan oleh pemrogram sendiri, dan yang lainnya dilakukan oleh penganalisis sistem.

7. Mengimplementasikan dan mengevaluasi sistem. Di tahap terakhir dari pengembangan sistem, penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan sistem. Sebagian pelatihan tersebut dilakukan oleh vendor,

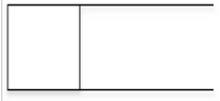
namun kesalahan pelatihan merupakan tanggung jawab penganalisis sistem.
[3]

2.5. Alat-alat Pengembangan Sistem

2.5.1. DFD (Data Flow Diagram)

Pengertian DFD secara umum adalah suatu network yang menggambarkan suatu sistem komputerisasi, manualisasi atau gabungan, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berkaitan sesuai dengan aturan yang sudah ada. DFD memiliki kelebihan dan kekurangan, kelebihannya adalah memungkinkan untuk menggambarkan sistem dari level yang tinggi lalu menguraikannya ke level yang lebih rendah, sedangkan kekurangannya adalah tidak menunjukkan proses looping, proses keputusan dan proses perhitungan.

Tabel 2. 1 Tabel DFD (Data Flow Diagram)

NO	Simbol	Nama Simbol	Arti Simbol
1		<i>External entity</i>	Simbol ini digunakan untuk menggambarkan asal ataupun tujuan data
2		Proses	Simbol proses digunakan untuk proses pengolahan atau transformasi data
3		<i>Data flow</i>	Simbol ini digunakan untuk menggambarkan aliran data berjalan
4		<i>Data Store</i>	Simbol ini digunakan untuk menggambarkan <i>data store</i> yang sudah disimpan

Keterangan simbol- simbol tersebut adalah sebagai berikut [3]:

1. Kesatuan luar (*external entity*)

Kesatuan luar adalah elemen-elemen lingkungan yang berada di luar batas sistem. Dimana elemen luar ini menyediakan suatu masukan (*input*) dan menerima keluaran (*output*) pada sistem. Nama entitas digunakan untuk menggabungkan elemen lingkungan yang menandai titik-titik berakhirnya sistem. Entitas ini digambarkan dalam DFD dengan suatu kotak atau segi empat dimana simbol entitas diberi label dengan nama lingkungannya.

2. Arus data (*data flow*)

Data flow terdiri atas sekelompok elemen data yang berhubungan secara logis yang bergerak dari satu titik ke titik lain yaitu ada panah yang berfungsi untuk menggambarkan arus tersebut. Panah itu digambarkan biasanya garis lurus ataupun garis melengkung.

3. Proses (*process*)

Proses adalah dimana suatu masukan (*input*) diubah menjadi keluaran (*output*). Proses ini digambarkan dengan lingkaransegi empat horizontal (segi empat tegak dengan sudut-sudut yang membulat) dimana setiap simbol proses diidentifikasi dengan label.

4. Penyimpanan data (*data store*)

Penyimpanan data adalah tempat dimana suatu data perlu dipertahankan dengan suatu alasan. Pada DFD, penyimpanan data adalah suatu tempat penampungan data. Cara penggambaran *data store* adalah satu set garis paralel, segi empat terbuka atau berbentuk lonjong.

Pendekatan aliran data memiliki 4 kelebihan utama yaitu [3]:

1. Kebebasan menjalankan implementasi teknis sistem yang terlalu dini (muda)
2. Pemahaman lebih jauh tentang keterkaitan (hubungan) satu sama yang lain dalam sistem dan sub sistem
3. Mengkomunikasikan pengetahuan sistem yang ada pada pengguna melalui DFD (*data flow diagram*)
4. Menganalisis sistem yang diajukan untuk menentukan apakah data dan proses yang dibutuhkan sudah ditetapkan.

Berikut adalah langkah-langkah membuat *data flow diagram (DFD)* [3]:

1. Menciptakan diagram konteks

Diagram konteks adalah tingkatan tertinggi dalam diagram aliran data dan hanya berisi satu proses dan dapat menunjukkan sistem secara keseluruhan. Pada diagram konteks proses tersebut diberi nomor nol (0), semua entitas eksternal diarahkan dari sistem dan menuju sistem dengan menggunakan aliran data. Pembuatan DFD berasal dari hasil penganalisis yang melakukan wawancara ataupun observasi langsung sebagai hasil analisis dokumen.

2. Menggambarkan diagram level 0 (level berikutnya)

Level ini bisa dicapai dengan cara mengembangkan diagram. Masukan (*input*) dan keluaran (*output*) yang ditetapkan dalam diagram pertama harus tetap konstan dalam semua diagram sub-urutannya. Sisa diagram asli dikembangkan kedalam gambaran terperinci yang melibatkan tiga sampai sembilan proses yang menunjukkan penyimpanan data dan aliran-aliran data baru di level yang lebih rendah. Diagram 0 adalah diagram dimana pengembangan diagram konteks bisa mencakup sampai sembilan proses jika memasukkan proses yang lebih banyak akan menyebabkan diagram yang susah dipahami.

Menciptakan diagram anak tingkat (tingkat yang lebih mendetail) Setiap proses pada diagram 0 bisa dikembangkan yang berguna untuk menciptakan diagram anak yang lebih mendetail. Diagram 0 ada diagram induk dan diagram anak. Diagram induk (*Parent process*) menghasilkan diagram anak (*child diagram*). Didalam menciptakan diagram anak ada aturan yaitu diagram anak tidak bisa menghasilkan keluaran ataupun menerima masukan dimana proses induknya juga tidak bisa menghasilkan atau menerima. Semua aliran data yang menuju atau keluar dari proses induk harus ditunjukkan mengalir kedalam atau keluar dari diagram anak tersebut. [3]

2.5.2. Kamus Data

Kamus data merupakan suatu aplikasi khusus dari kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Kamus data merupakan hasil

referensi data mengenai data, suatu data yang disusun oleh penganalisis sistem sebagai bimbingan selama melakukan analisis dan desain.

Ketentuan-ketentuan dalam yang digunakan dalam kamus data, serta bagaimana kamus data dikembangkan adalah hal-hal yang tetap berhubungan dengan penganalisis sistem. Memahami proses penyusunan suatu kamus data bisa membantu penganalisis sistem mengkonseptualisasikan serta mengurangi redundansi, kamus data bisa digunakan untuk:

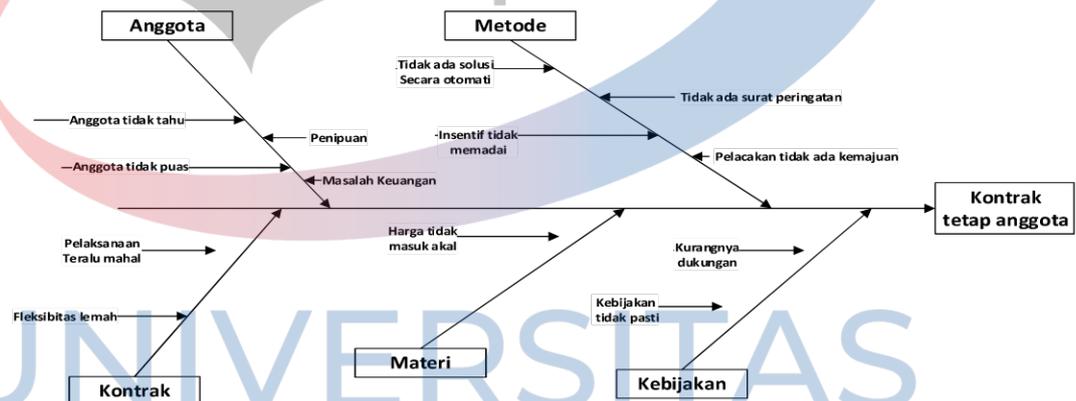
1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam file-file

Struktur data biasanya digambarkan menggunakan notasi aljabar. Metode ini memungkinkan penganalisis membuat suatu gambaran mengenai elemen-elemen yang membentuk struktur data bersama-sama dengan informasi-informasi mengenai elemen-elemen tersebut. Notasi aljabar menggunakan simbol-simbol sebagai berikut [3]:

1. Tanda sama dengan (=) artinya “Terdiri dari”
2. Tanda plus (+) artinya “Dan”
3. Tanda kurung {} menunjukkan elemen-elemen repetitif, juga disebut dengan kelompok berulang atau tabel-tabel. Kemudian bisa ada satu atau beberapa elemen berulang didalam kelompok tersebut.
4. Tanda kurung [] menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada, sedangkan elemen lainnya juga ada. Tetapi tidak bisa kedua-duanya ada secara bersamaan. Elemen-elemen yang ada didalam tanda kurung ini saling terpisah satu sama lain.
5. Tanda kurung () menunjukkan satu elemen yang bersifat pilihan. Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga memuat spasi atau nol untuk field-field numeric pada struktur file.

2.5.3. Diagram Fishbone

Proses untuk menganalisis, mengidentifikasi, dan menyelesaikan masalah sering disebut dengan diagram ishikawa atau diagram *fishbone*. Konsep diagram fishbone adalah nama masalah yang mendapat perhatian di cantumkan disebelah kanan diagram (atau pada kepala ikan) dan penyebab masalah digambarkan sebagai tulang-tulang dari tulang utama. Secara khusus tulang-tulang ini mendeskripsikan empat kategori dasar yaitu material, mesin, kekuatan manusia dan metode (empat M : *material, machine, manpower, method*). Kategori alternatif atau tambahan meliputi tempat, prosedur, kebijakan, dan orang atau disebut empat P yang meliputi (*place, procedure, policy, people*) atau lingkungan sekeliling, pemasok, sistem dan keterampilan atau disebut empat S yang meliputi (*surrounding, supplier, system, skill*) [4]



Gambar 2. 5 Diagram Fishbone

2.5.4. Kerangka PIECES

Proses dan teknik yang digunakan untuk analisis sistem untuk mengidentifikasi, menganalisis, dan memahami persyaratan sistem disebut *requirements discovery* / penemuan persyaratan. Penemuan persyaratan melibatkan analisis sistem yang bekerjasama dengan pengguna dan pemilik sistem selama fase pengembangan sistem mula-mula untuk mendapatkan pemahaman yang rinci mengenai persyaratan bisnis dari sistem informasi.

Persyaratan sistem menentukan apa yang seharusnya dikerjakan oleh sistem informasi atau properti serta kualitas apa yang harus dimiliki oleh sistem.

Persyaratan sistem yang menetapkan apa yang harusnya dilakukan oleh sistem informasi sering disebut persyaratan fungsional. Persyaratan sistem yang menetapkan properti atau kualitas yang harus dimiliki oleh sistem sering disebut persyaratan fungsional.

Kerangka PIECES memberikan alat unggul untuk menggolongkan persyaratan sistem. Keuntungan menggolongkan berbagai tipe persyaratan adalah kemampuan untuk menggolongkan persyaratan tersebut untuk tujuan pelaporan, pelacakan, dan validasi. Hal tersebut membantu identifikasi persyaratan sistem secara cermat. Kategori-kategori kerangka PIECES adalah [4]:

1. P : Kebutuhan untuk mengoreksi atau memperbaiki *Performance*/Performa
2. I : Kebutuhan untuk mengoreksi atau memperbaiki Informasi
3. E : Kebutuhan untuk mengoreksi atau memperbaiki *Ekonomi*/Ekonomi mengendalikan biaya, atau meningkatkan keuntungan.
4. C : Kebutuhan untuk mengoreksi atau memperbaiki *Control*/Kontrol atau keamanan
5. E : Kebutuhan untuk mengoreksi atau memperbaiki *Efficiency*/Efisiensi orang dan proses
6. S : Kebutuhan untuk mengoreksi atau memperbaiki *Service*/Layanan kepada member.

2.5.5. Use Case Diagram

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya *Login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use Case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *Use Case* dapat meng-*include* fungsionalitas *Use Case* lain sebagai bagian dari proses dalam dirinya. Secara

umum diasumsikan bahwa *Use Case* yang di-include akan dipanggil setiap kali *Use Case* yang meng-include dieksekusi secara normal. Sebuah *Use Case* dapat di-include oleh lebih dari satu *Use Case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *Use Case* juga dapat meng-extend *Use Case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *Use Case* menunjukkan bahwa *Use Case* yang satu merupakan spesialisasi dari yang lain. Diagram *Use Case* adalah merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing – masing diagram *Use Case* menunjukkan sekumpulan *Use Case*, aktor dan hubungannya. Diagram *Use Case* adalah penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram *Use Case* merupakan pusat pemodelan perilaku sistem, subsistem dan kelas. *Use Case* adalah interaksi antara aktor eksternal dan sistem, hasil yang dapat diamati oleh aktor, berorientasi pada tujuan, dideskripsikan di diagram *Use Case* dan teks. Diagram *Use Case* melibatkan: [4]

Tabel 2. 2 Use Case Diagram

Nama Simbol	Simbol
<i>Actor</i>	
<i>Use Case</i>	
<i>Association Relationship</i>	
<i>Include Relationship</i>	
<i>Extend Relationship</i>	
<i>Generalisasi Relationship</i>	

1. Sistem yaitu sesuatu yang hendak kita bangun

2. Aktor adalah entitas – entitas luar yang berkomunikasi dengan sistem
3. *Use Case* adalah fungsionalitas yang di persepsi oleh aktor
4. Relasi adalah relasi antara aktor dengan *Use Case*.



UNIVERSITAS MIKROSKIL