

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Konsep Sistem Informasi

Konsep sistem mendasari semua proses bisnis, serta pemahaman tentang sistem informasi dan teknologi. Sistem informasi terdiri dari sistem dan informasi. Sistem merupakan sebuah metode untuk mencapai sebuah tujuan. Sedangkan informasi merupakan hasil dari pengolahan data. Maka sistem informasi digunakan untuk mendukung fungsi sebuah organisasi yang sifatnya manajerial [2]. Sistem informasi yang ideal harus bisa menyajikan data valid yang bisa di akses kapan saja informasi itu dibutuhkan. Di dalam konsep dasar sistem informasi terdapat tiga aktivitas dasar yaitu [2]:

1. *Input*, merupakan data mentah yang dimasukkan ke dalam sistem informasi untuk diproses.
2. *Process*, input atau data yang dimasukkan ke dalam sistem akan diproses dengan prosedur tertentu untuk menghasilkan *output* yang sesuai.
3. *Output*, setelah *input* selesai diproses, maka sistem akan menghasilkan *output* yang diinginkan oleh pemakai sistem. *Output* yang dikeluarkan berupa informasi yang dapat membantu dan bermanfaat bagi pemakai sistem.

##### 2.1.1 Sistem

Kata sistem berasal dari bahasa Yunani, yaitu *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang independen, namun saling terkait sebagai satu kesatuan. Sistem terdiri atas struktur dan proses. Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut, sedangkan proses sistem menjelaskan cara kerja setiap unsur sistem dalam mencapai tujuan [3]. Setiap sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri atas berbagai sistem yang lebih kecil, yang disebut subsistem. Setiap sistem diciptakan untuk menangani sesuatu yang berulang-ulang atau yang secara rutin terjadi. Jadi sistem bisa didefinisikan menjadi seperangkat elemen independen, tetapi saling

berhubungan sebagai satu kesatuan [3]. Suatu sistem mempunyai ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam mengidentifikasi pembuatan sistem. Adapun karakteristik sistem yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Component*), sistem terdiri atas berbagai komponen yang saling berinteraksi dan bekerja sama.
2. Batasan Sistem (*Boundary*), daerah batasan sistem yang satu dengan lainnya atau dengan lingkungan kerjanya.
3. Lingkungan Luar Sistem (*Environment*), sistem yang beraktivitas dan berinteraksi satu sama lain demi mencapai tujuan sesuai sasarannya.
4. Penghubung Sistem (*Interface*), media penghubung antar subsistem, media ini memungkinkan pengaliran sumber daya dari satu subsistem ke subsistem lainnya.
5. Masukan Sistem (*Input*), energi yang masuk kedalam sebuah sistem, seperti *maintenance input* dan *signal input*.
6. Keluaran Sistem (*Output*), energi yang telah diolah serta diklasifikasikan sebagai keluaran yang berguna dan sisa pembuangan [4].

### 2.1.2 Informasi

Informasi adalah kumpulan data yang diorganisasikan dan diolah sehingga memiliki nilai tambah melebihi nilai fakta individu. Sumber dari informasi adalah data. Data menggambarkan suatu kejadian yang benar-benar terjadi pada saat tertentu [3]. Data diolah melalui suatu metode untuk menghasilkan informasi. Informasi yang dihasilkan digunakan untuk membuat suatu keputusan. Informasi juga memiliki kualitas yaitu:

1. Akurat, informasi yang kebenarannya dipertanggungjawabkan berdasarkan bukti-bukti yang memadai sehingga informasi yang disampaikan memiliki keakuratan dan tidak diragukan kebenarannya.
2. Tepat Pada Waktunya, informasi harus tersedia pada saat informasi diperlukan, karena bila informasi terlambat maka informasi itu tidak memiliki nilai yang baik lagi dan tidak bisa digunakan untuk mengambil keputusan dan hal itu bisa berakibat sangat fatal.

3. Relevan, informasi harus sesuai dengan apa yang dibutuhkan. Informasi juga memiliki tingkat relativitas yang berbeda, tergantung pada tingkatannya [3]. Nilai dari informasi bisa ditentukan dari dua hal, yaitu manfaat dan biaya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*.

### 2.1.3 Sistem Informasi

Sistem informasi merupakan sekumpulan subsistem yang membentuk suatu komponen yang bertujuan untuk menghasilkan informasi. Sistem informasi juga melakukan pengklasifikasian arus informasi, hal ini dikarenakan adanya keragaman kebutuhan informasi para pengguna [5]. Dalam mengelola informasi dibutuhkan subsistem yang saling berhubungan sehingga membentuk suatu komponen yang meliputi proses *input/output*. Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yaitu:

1. Blok Masukan (*Input Block*), metode dan media yang digunakan untuk menangkap dokumen dasar yang masuk ke dalam sistem informasi.
2. Blok Model (*Model Block*), memanipulasi data *input* dan data yang tersimpan di basis data yang berisikan kombinasi prosedur, logika dan metode matematik untuk menghasilkan *output* yang diinginkan.
3. Blok Keluaran (*Output Block*), keluaran yang memiliki kualitas informasi dan dokumentasi yang berguna untuk semua pemakai sistem dan tingkatan manajemen.
4. Blok Teknologi (*Technologi Block*), menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan membantu pengendalian diri secara keseluruhan.
5. Blok Basis Data (*Database Block*), kumpulan dari beberapa data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. Blok Kendali (*Control Block*), pencegahan terhadap kerusakan sistem dan merancang solusi apabila terjadi kesalahan sistem.

## 2.2 Website

*Website* merupakan media yang dipakai untuk menampung sebuah data teks, gambar, animasi, dan suara yang dapat ditampilkan di internet dan dapat diakses oleh komputer lain yang terhubung dengan internet [7]. *Website* berisikan sekumpulan halaman yang disimpan dalam hosting dan memiliki nama domain atau sub-domain sehingga bisa di akses melalui internet [7].

Biasanya *website* bisa digunakan untuk:

1. Membuat pengumuman atau pemberitahuan.
2. Memberi informasi dan layanan jasa yang diberikan oleh suatu instansi atau perusahaan.
3. *Update* atau pembaruan informasi secara cepat sesuai dengan perkembangan dunia teknologi informasi [6].

Dalam membuat situs *website* memerlukan hal berikut:

1. *Text Editor*, perangkat lunak yang digunakan untuk menulis kode. Seperti Notepad++ dan Visual Studio Code.
2. *Upload/Publishing Tool*, platform yang cocok digunakan untuk melakukan publikasi seperti Blog dan Adobe Dreamweaver.
3. *Web Hosting Service*, layanan yang digunakan untuk menerbitkan *website* secara daring. *Web hosting service* juga bertugas untuk menjaga *server* agar tetap aktif dan berjalan, mengamankan dari serangan *cyber* berbahaya, dan memindahkan konteks (teks, gambar, *file*) dari *server* ke *browser* pengunjung seperti Cloud Hosting dan WordPress Hosting.
4. *Browser*, perangkat lunak yang digunakan untuk membuka halaman situs internet seperti Google Chrome, Internet Explorer, Microsoft Edge dan Firefox [7].

### 2.2.1 User Interface

*User interface* adalah representasi visual dari produk digital. UI merupakan penghubung antara pengguna dan fungsionalitas suatu produk. *Interface* berisikan tata letak, tipografi, warna, animasi, gambar, teks, grafik, dan interaksi mikro. Arsitektur informasi yang ditentukan UX adalah *blueprint* untuk tampilan akhir UI yang disempurnakan [8]. UI juga memiliki *framework* yang berfungsi untuk

menyediakan pustaka komponen seperti Kit UI, Toolkit UI, *Framework* CSS dan *Frontend Frameworks*. Contoh *Framework* UI : Bootstrap, Foundation, Sematic UI, *Materialize*, *Blueprint* dan Ulkit [9]. UI *designer* bertanggung jawab atas tampilan dan nuansa akhir produk. Tugasnya yaitu menentukan gaya unik dan membuat tampilan UI sesuai dengan target pasar [8]. Representasi visual harus dapat dibaca, dapat digunakan, dan tanpa elemen yang mengganggu dan tidak perlu. Desain yang baik adalah:

1. Desain yang baik adalah inovatif, perkembangan teknologi menawarkan peluang baru untuk desain yang inovatif.
2. Desain yang baik membuat produk menjadi berguna, suatu produk yang dibeli harus bisa digunakan. Dan hal itu harus memenuhi kriteria tertentu, tidak hanya fungsional, tetapi juga psikologis dan estetika.
3. Desain yang baik adalah memiliki estetika, kualitas estetika suatu produk merupakan bagian integral dari kegunaannya karena produk yang digunakan setiap hari mempengaruhi pribadi dan kesejahteraan. Tetapi hanya objek yang dieksekusi dengan baik yang bisa menjadi indah.
4. Desain yang bagus tahan lama, jika menemukan *grid* dan *whitespace* maka desain akan tetap bermanfaat, bahkan setelah perubahan besar dalam tren desain. Mungkin gradien atau ilustrasinya yang tidak akan tahan lama, tetapi presisi dan kejelasan desain tidak akan menua.
5. Desain yang baik membuat produk dapat dimengerti, UI yang dapat dimengerti adalah perpaduan antara *readability* yang bagus, *hierarchy* yang tepat dan jenis konten yang tepat. Hal ini berasal dari pendekatan yang tepat terhadap tata letak, dan presisi keseleruhan.
6. Desain yang bagus tidak mencolok, desain yang dibuat harus netral dan terkendali. Karena peran UI adalah untuk menyampaikan konten, jadi tidak perlu menggunakan terlalu banyak animasi dan warna yang mencolok.
7. Desain yang baik adalah jujur, memanipulasi konsumen dengan janji-janji yang tidak dapat ditepati seperti desain yang dibuat terlalu rumit seperti bagan dan grafik.

8. Desain yang baik adalah menyeluruh sampai ke detail terakhir, desain yang tepat dapat dilihat terutama dengan elemen yang tidak terlalu penting yang masih dibuat dengan hati-hati dan detail.
9. Desain yang baik adalah ramah lingkungan, UI mempengaruhi lingkungan dalam beberapa cara. UI dapat menghemat sumber daya dan meminimalkan polusi fisik dan visual di seluruh siklus hidup produk. Seperti mengoptimalkan aplikasi untuk memuat lebih cepat.
10. Desain yang baik adalah desain yang sesedikit mungkin, karena desain akan lebih berkonsentrasi pada aspek-aspek penting, dan produk tidak dibebani dengan hal-hal yang tidak penting. UI dibuat untuk mengirimkan konten (teks, suara, video, gambar, dll.). Ornamen dan elemen yang tidak perlu dapat menyebabkan proyek yang dirancang berlebihan yang berdampak negatif pada pengalaman aktual [8].

### 2.2.2 User Experience (UX)

*User experience* merupakan interaksi antara pengguna manusia, komputer dan produk berbasis komputer, seperti situs web, aplikasi, dan sistem. Dengan bantuan *UX designer* kita bisa menggambarkan bagaimana perasaan orang ketika mereka menggunakan produk atau layanan yang ditawarkan. Kemudian *UX designer* menerapkan pengetahuan ke pengembangan produk untuk memastikan bahwa pengguna memiliki pengalaman terbaik dengan suatu produk [10].

Dalam merancang sebuah UX, *UX designer* harus memperhatikan kegunaan dari UX karena apabila kegunaan tidak dapat mencapai tujuan secara efisien, efektif dan dengan cara yang memuaskan. Apalagi UX pada *website*, jika produk tidak dapat digunakan, maka UX-nya akan buruk. Pengguna situs web menunjukkan bahwa 46% pengguna meninggalkan situs web karena kurangnya pengiriman pesan terhadap *website*, 44% pengguna keluar karena kurangnya informasi kontak, dan 37% pengguna pergi karena desain atau navigasi yang buruk [10]. Kunci utama dalam membuat UX:

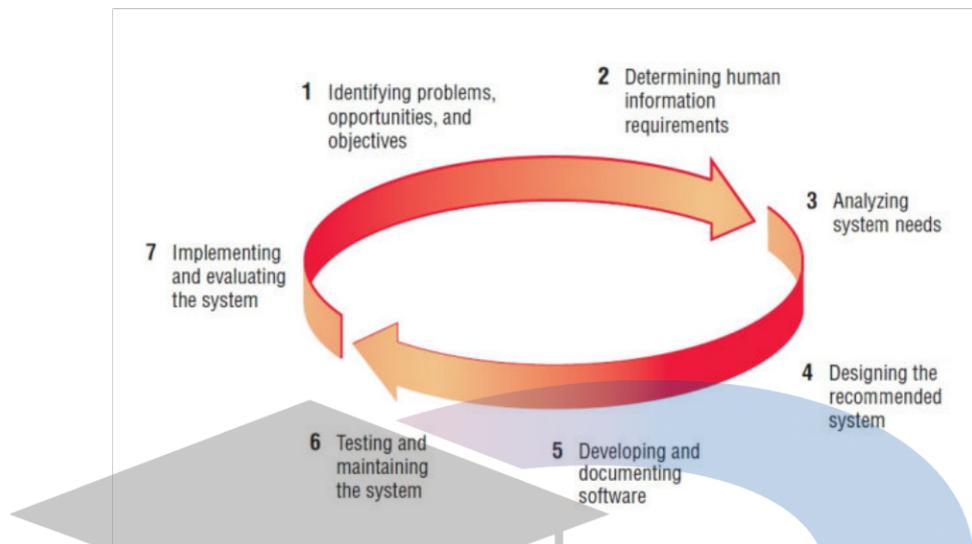
1. Desain, apa itu desain dan bagaimana cara melakukannya?
2. Teknologi, sistem interaktif, produk, perangkat, dan komponen itu sendiri.

3. Orang, perancang UX perlu mempertimbangkan siapa yang akan menggunakan sistem dan layanan.
4. Aktivitas dan konteks, UX adalah tentang apa yang ingin dilakukan orang, tentang tujuan, perasaan, dan pencapaian mereka. UX perlu mempertimbangkan konteks dimana aktivitas tersebut berlangsung [11].

### 2.3 System Development Life Cycle (SDLC)

*System Development Life Cycle* (SDLC) adalah pendekatan yang dilakukan secara bertahap dalam hal melakukan analisa dan membangun rancangan sistem dengan menggunakan siklus-siklus secara spesifik terhadap kegiatan penggunaannya. Beberapa analis tidak setuju tentang berapa banyak fase yang ada, tetapi mereka juga memuji pendekatan terorganisirnya. Dalam pendekatan ini telah dibagi siklus menjadi tujuh fase. Meskipun setiap fase disajikan secara terpisah, hal itu tidak pernah dicapai sebagai langkah terpisah. Sebaliknya, beberapa kegiatan dapat terjadi secara bersamaan, dan kegiatan dapat diulang.

1. Mengidentifikasi masalah, peluang dan tujuan.
2. Menentukan syarat-syarat informasi.
3. Menganalisis kebutuhan sistem.
4. Merancang sistem yang direkomendasikan.
5. Mengembangkan dan mendokumentasikan perangkat lunak.
6. Menguji dan mempertahankan sistem.
7. Menerapkan dan mengevaluasi sistem [1].



Gambar 2. 1 Fase pada *Systems Development Life Cycle (SDLC)*

#### 1. Mengidentifikasi Masalah, Peluang, dan Tujuan

Mengidentifikasi masalah, peluang, dan tujuan sangatlah penting dalam keberhasilan sebuah proyek. Karena tidak ada yang mau menyalahkan waktu berikutnya untuk mengatasi masalah yang salah. Fase pertama mengharuskan analis untuk melihat dengan jujur apa yang terjadi dalam bisnis. Kemudian, bersama dengan anggota organisasi lainnya, analis menunjukkan masalah. Sering kali orang lain akan memunculkan masalah ini, dan mereka adalah alasan analis awalnya dipanggil. Peluang adalah situasi yang diyakini analis dapat ditingkatkan melalui penggunaan sistem informasi yang terkomputerisasi. Memanfaatkan peluang dapat memungkinkan bisnis untuk mendapatkan keunggulan kompetitif atau menetapkan standar industri. Identifikasi tujuan juga merupakan komponen penting dari fase pertama. Analisis pertama-tama harus menemukan apa yang coba dilakukan oleh bisnis. Kemudian analis akan dapat melihat apakah beberapa aspek aplikasi sistem informasi dapat membantu bisnis mencapai tujuannya dengan mengatasi masalah atau peluang tertentu. Orang-orang yang terlibat dalam fase pertama adalah para pengguna, analis, dan manajer sistem yang mengoordinasikan proyek. Kegiatan dalam fase ini terdiri dari mewawancarai manajemen pengguna, meringkas pengetahuan yang diperoleh, memperkirakan ruang lingkup proyek, dan mendokumentasikan hasilnya. *Output* dari fase ini adalah

laporan kelayakan yang berisi definisi masalah dan meringkas tujuan. Manajemen kemudian harus membuat keputusan apakah akan melanjutkan proyek yang diusulkan. Jika grup pengguna tidak memiliki dana yang cukup dalam anggarannya atau ingin mengatasi masalah yang tidak terkait, atau jika masalah tidak memerlukan sistem komputer, solusi yang berbeda mungkin disarankan, dan proyek sistem tidak dilanjutkan lebih jauh.

## 2. Menentukan Syarat-Syarat Informasi

Fase selanjutnya yang dimasukkan analisis adalah menentukan kebutuhan manusia pengguna yang terlibat, menggunakan berbagai alat untuk memahami bagaimana pengguna berinteraksi dalam konteks kerja dengan sistem informasi mereka saat ini. Analisis akan menggunakan metode interaktif seperti wawancara, pengambilan sampel dan investigasi data keras, dan kuesioner, bersama dengan metode yang tidak mengganggu, seperti mengamati perilaku pengambil keputusan dan lingkungan kantor mereka, dan metode yang mencakup semua, seperti *prototyping*. Analisis akan menggunakan metode ini untuk mengajukan dan menjawab banyak pertanyaan tentang *human computer interaction* (HCI), termasuk pertanyaan seperti, "Apa kekuatan dan keterbatasan fisik pengguna?" Dengan kata lain, "Apa yang perlu dilakukan untuk membuat sistem terdengar, dapat dibaca, dan aman?" "Bagaimana sistem yang baru dirancang agar mudah digunakan, dipelajari, dan diingat?" "Bagaimana sistem bisa dibuat menyenangkan atau bahkan menyenangkan untuk digunakan?" "Bagaimana sistem dapat mendukung suatu tugas pekerjaan individu pengguna dan membuatnya lebih produktif dengan cara baru?".

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

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

### 3. Menganalisis Kebutuhan Sistem

Fase berikutnya yang dilakukan oleh analisis sistem melibatkan analisis kebutuhan sistem. Sekali lagi, alat dan teknik khusus membantu analisis membuat penentuan kebutuhan. Alat-alat seperti *data flow diagram* (DFD) untuk memetakan *input*, *process*, dan *output* fungsi bisnis, atau diagram aktivitas atau diagram urutan untuk menunjukkan urutan peristiwa, menggambarkan sistem dalam bentuk grafis terstruktur. Dari aliran data, urutan, atau diagram lain, kamus data dikembangkan yang mencantumkan semua item data yang digunakan dalam sistem, serta spesifikasinya. Selama fase ini analisis sistem juga menganalisis keputusan terstruktur yang dibuat. Keputusan terstruktur adalah keputusan yang menentukan kondisi, alternatif kondisi, tindakan, dan aturan tindakan. Ada tiga metode utama untuk analisis keputusan terstruktur: bahasa Inggris terstruktur, tabel keputusan, dan pohon keputusan. Pada titik ini di SDLC, analisis sistem menyiapkan proposal sistem yang merangkum apa yang telah diketahui tentang pengguna, kegunaan, dan kegunaan sistem saat ini; memberikan analisis biaya-manfaat dari alternatif; dan membuat rekomendasi tentang apa (jika ada) yang harus dilakukan. Jika salah satu rekomendasi dapat diterima oleh manajemen, analisis melanjutkannya. Setiap masalah sistem adalah unik, dan tidak pernah ada satu solusi yang benar. Cara dimana rekomendasi atau solusi dirumuskan tergantung pada kualitas individu dan pelatihan profesional masing-masing analisis dan interaksi analisis dengan pengguna dalam konteks lingkungan kerja mereka.

### 4. Merancang Sistem yang Direkomendasikan

Dalam fase desain SDLC, analis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk mencapai desain logis dari sistem informasi. Analis merancang prosedur bagi pengguna untuk membantu mereka memasukkan data secara akurat sehingga data yang masuk ke sistem informasi benar. Selain itu, analis menyediakan bagi pengguna untuk menyelesaikan *input* efektif ke sistem informasi dengan menggunakan teknik bentuk yang baik dan halaman *website* atau desain layar. Bagian dari desain logis dari sistem informasi adalah merancang HCI. Antarmuka menghubungkan pengguna dengan sistem dan dengan demikian sangat penting. Antarmuka pengguna dirancang dengan bantuan pengguna untuk memastikan bahwa sistemnya dapat didengar, terbaca, dan aman, serta menarik dan menyenangkan untuk digunakan. Contoh antarmuka pengguna fisik termasuk *keyboard* (untuk mengetik pertanyaan dan jawaban), menu pada layar (untuk memperoleh perintah pengguna), dan berbagai antarmuka pengguna grafis (GUI) yang menggunakan mouse atau layar sentuh. Fase desain juga mencakup perancangan basis data yang akan menyimpan banyak data yang dibutuhkan oleh pembuat keputusan dalam organisasi. Pengguna mendapat manfaat dari *database* yang terorganisir dengan baik yang logis bagi mereka dan sesuai dengan cara mereka melihat pekerjaan mereka. Dalam fase ini analis juga bekerja dengan pengguna untuk mendesain *output* (baik pada layar atau dicetak) yang memenuhi kebutuhan informasi mereka. Akhirnya, analis harus merancang kontrol dan prosedur cadangan untuk melindungi sistem dan data, dan untuk menghasilkan paket spesifikasi program untuk programmer. Setiap paket harus berisi tata letak *input* dan *output*, spesifikasi *file*, dan detail pemrosesan; itu juga dapat mencakup pohon keputusan atau tabel, UML atau diagram alir data, dan nama dan fungsi dari setiap kode yang ditulis sebelumnya baik yang ditulis sendiri atau menggunakan kode atau pustaka kelas lainnya.

##### 5. Mengembangkan dan Mendokumentasikan Perangkat Lunak

Pada fase kelima SDLC, analis bekerja dengan programmer untuk mengembangkan perangkat lunak asli yang diperlukan. Selama fase ini analis bekerja dengan pengguna untuk mengembangkan dokumentasi yang efektif

untuk perangkat lunak, termasuk manual prosedur, bantuan daring, dan situs *website* yang menampilkan *Frequently Asked Questions* (FAQ), pada *file* ReadMe yang dikirimkan dengan perangkat lunak baru. Karena pengguna terlibat sejak awal, dokumentasi fase harus menjawab pertanyaan yang mereka ajukan dan selesaikan bersama dengan analis. Dokumentasi memberi tahu pengguna cara menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi masalah perangkat lunak. Programmer menjadi peran kunci dalam fase ini karena mereka merancang, kode dan menghapus kesalahan sintaks dari program komputer. Untuk memastikan kualitas, seorang programmer dapat melakukan desain atau kode berjalan, menjelaskan bagian-bagian kompleks dari program kepada tim programmer lain.

#### 6. Menguji dan Mempertahankan Sistem

Sebelum sistem informasi dapat digunakan, itu harus diuji. Hal ini jauh lebih murah untuk menangkap masalah sebelum sistem digunakan pengguna. Beberapa pengujian diselesaikan oleh pemrogram saja, sebagian lagi oleh analis sistem bersama dengan pemrogram. Pemeriksaan tes untuk menunjukkan masalah dijalankan pertama dengan data sampel dan akhirnya dengan data aktual dari sistem saat ini. Seringkali rencana pengujian dibuat di awal SDLC dan disempurnakan seiring proyek berlangsung. Pemeliharaan sistem dan dokumentasinya dimulai pada fase ini dan dilakukan secara rutin sepanjang umur sistem informasi. Sebagian besar pekerjaan rutin programmer terdiri dari pemeliharaan, dan bisnis menghabiskan banyak uang untuk pemeliharaan. Beberapa pemeliharaan, seperti pembaruan program, dapat dilakukan secara otomatis melalui situs vendor di *website*. Banyak prosedur sistematis yang digunakan analis di seluruh SDLC dapat membantu memastikan bahwa pemeliharaan dijaga agar tetap minimum.

#### 7. Mengimplementasikan dan Mengevaluasi Sistem.

Dalam fase terakhir pengembangan sistem ini, analis membantu mengimplementasikan sistem informasi. Fase ini melibatkan melatih pengguna untuk menangani sistem. Vendor melakukan beberapa pelatihan, tetapi pengawasan pelatihan adalah tanggung jawab analis sistem. Selain itu, analis

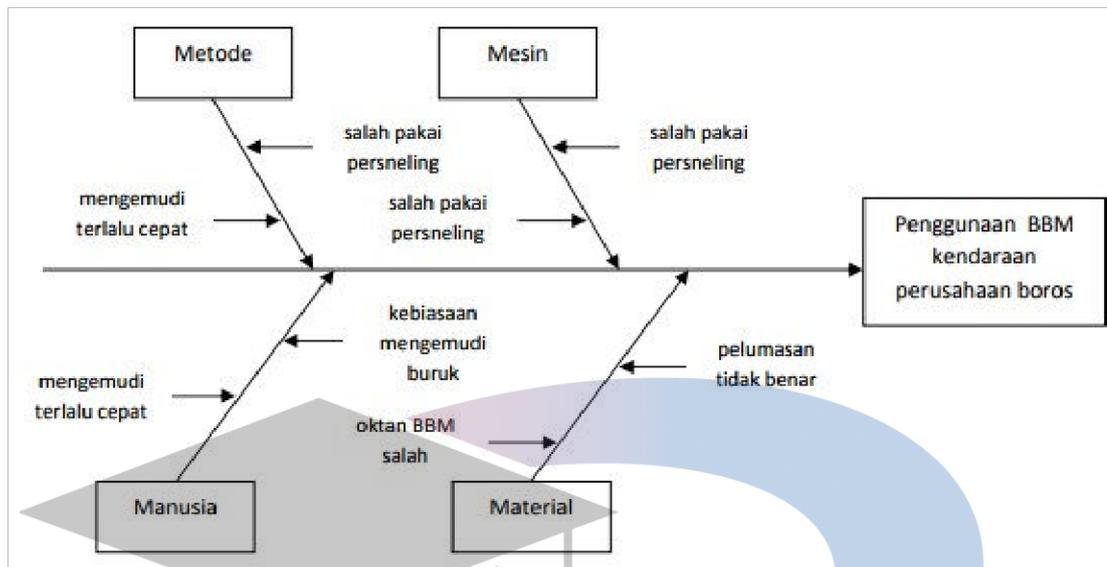
perlu merencanakan konversi yang lancar dari sistem lama ke yang baru. Proses ini termasuk mengkonversi *file* dari format lama ke yang baru, atau membangun basis data, memasang peralatan, dan membawa sistem baru ke dalam produksi. Evaluasi dimasukkan sebagai bagian dari fase akhir SDLC ini. Sebenarnya, evaluasi berlangsung selama setiap fase. Kriteria utama yang harus dipenuhi adalah apakah pengguna yang dituju memang menggunakan sistem. Perlu dicatat bahwa kerja sistem sering kali bersifat siklis. Ketika seorang analis menyelesaikan satu fase pengembangan sistem dan melanjutkan ke tahap berikutnya, penemuan masalah dapat memaksa analis untuk kembali ke fase sebelumnya dan memodifikasi pekerjaan yang dilakukan disana [1].

## 2.4 Alat Bantu Dalam Perancangan Sistem

Dalam melakukan perancangan sistem informasi, dibutuhkan beberapa alat bantu perancangan agar mempermudah dalam merancang suatu sistem dan dapat mencapai hasil yang maksimal.

### 2.4.1 Fishbone

*Fishbone* diagram (juga disebut diagram ishikawa atau diagram sebab-akibat) adalah teknik grafis untuk menunjukkan beberapa penyebab dari suatu peristiwa atau fenomena tertentu. Secara khusus, *fishbone* diagram (bentuknya mirip dengan kerangka ikan) adalah alat umum yang digunakan untuk analisis sebab dan akibat untuk mengidentifikasi interaksi kompleks penyebab untuk masalah atau peristiwa tertentu. Diagram kausal ini dibuat oleh Ishikawa tahun 1990 dalam bidang penelitian manajemen [12].



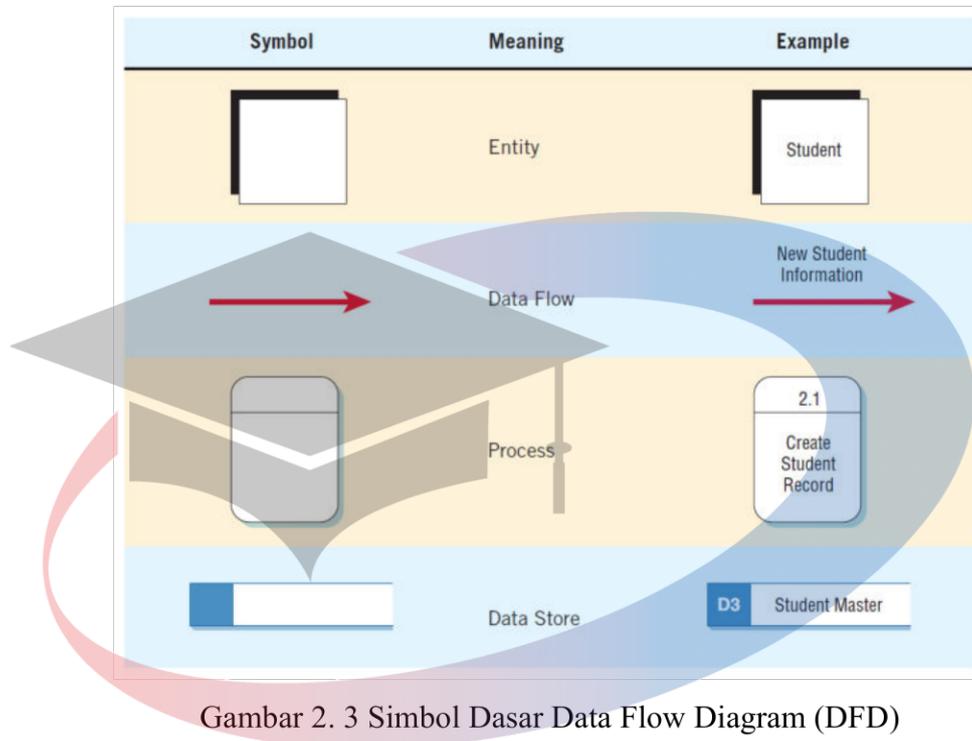
Gambar 2. 2 Contoh Ilustrasi *Fishbone*

Faktanya, analisis sebab akibat ini pada awalnya dikembangkan sebagai alat kontrol kualitas, seperti desain produk dan pencegahan cacat kualitas, untuk mengidentifikasi faktor-faktor potensial yang menyebabkan efek keseluruhan. Setiap penyebab merupakan sumber variasi dari fenomena yang diteliti. Penyebab biasanya dikelompokkan ke dalam kategori utama untuk mengidentifikasi keseluruhan sumber variasi yang mengarah pada efek utama. Secara umum, *fishbone* diagram dapat digunakan sebagai representasi visual yang tepat dari fenomena yang melibatkan penyelidikan berbagai faktor sebab-akibat dan bagaimana faktor-faktor tersebut saling terkait [1].

#### 2.4.2 Data Flow Diagram (DFD)

Ketika analisis sistem mencoba memahami kebutuhan informasi pengguna, mereka harus mampu mengkonseptualisasikan bagaimana data bergerak, proses atau transformasi yang dialami data, dan apa keluarannya. Salah satu perangkat yang dapat menggambarkan pemodelan sistem tersebut adalah *Data Flow Diagram* (DFD). Melalui teknik analisis terstruktur yang disebut DFD, seorang analis sistem dapat mengumpulkan representasi grafis dari proses data di seluruh organisasi. Dengan menggunakan kombinasi empat simbol, analis dapat membuat

penggambaran proses yang pada akhirnya akan memberikan dokumentasi sistem yang solid Gambar 2. 3 [1].



Gambar 2. 3 Simbol Dasar Data Flow Diagram (DFD)

Keunggulan dari DFD adalah mudah dipahami oleh orang teknik maupun non-teknik, memberikan gambaran sistem secara menyeluruh, lengkap dengan lingkup sistem dan hubungan sistem lainnya dan memberikan tampilan komponen-komponen sistem secara detail. DFD dapat dan harus digambar secara sistematis, seorang analis perlu mengkonseptualisasikan aliran data dari perspektif *top-down*.

Pedoman bagaimana menggambarkan DFD:

1. Identifikasi semua *external entity* yang terlibat dalam sistem.
2. Identifikasi semua *input* dan *output* yang terlibat dengan *external entity*.
3. Gambarkan *Context Diagram* berdasarkan *external entity*. dan *input*, *output* yang sudah diidentifikasi. Dalam suatu *Context Diagram* hanya mengandung satu dan hanya satu proses saja dan biasanya diberi nomor proses 0, proses ini mewakili dari seluruh sistem. *Context Diagram* akan digambarkan lebih rinci lagi yang disebut dengan *overview diagram*. Tiap-tiap proses di *overview diagram* akan digambar secara terinci dan di sebut DFD level 1. Tiap proses di level 1 akan digambarkan lebih terinci lagi dan disebut DFD level 2.

4. Gambarlah bagan berjenjang (*hierarchy chart*) untuk semua proses yang ada di sistem untuk mempersiapkan penggambaran DFD ke level berikutnya.
5. Gambarlah DFD untuk *overview diagram* (level 0) berdasarkan proses dibagan berjenjang.
6. Gambarlah DFD untuk level-level berikutnya, yaitu level 1, kemudian dipecah dalam level 2, dan seterusnya.
7. Gambarlah DFD level 2, yang merupakan penjabaran dari proses-proses yang ada di level 1 [13].

Aturan dalam membuat *Data Flow Diagram* (DFD):

1. Tidak boleh untuk menghubungkan *external entity*/kesatuan luar/terminator dengan *external entity* lainnya secara langsung.
2. Tidak boleh menghubungkan *data store*/simpanan data dengan *data store* secara langsung.
3. Tidak boleh menghubungkan *data store*/simpanan data dengan *external entity*/kesatuan luar secara langsung.
4. Pada setiap proses harus ada *data flow*/arus data yang masuk dan *data flow*/arus data yang keluar, demikian pula sebaliknya.
5. Tidak boleh ada proses dan arus data yang tidak memiliki nama, karena dapat mengakibatkan arus data yang tidak berhubungan bercampur.
6. Tidak boleh ada proses yang tidak memiliki nomor [5].

Metode membuat *Data Flow Diagram* (DFD):

1. Mulai yang umum sampai yang detail (*top-down-analisis*).
2. Jabarkan setiap proses sedetail mungkin.
3. Pelihara konsisten antar proses (perhatikan jumlah masukan ke proses dan keluar dari proses).
4. Berikan label atau nama yang bermakna.
  - a. Nama untuk *external entity*/kesatuan luar/terminator gunakan huruf besar.
  - b. Nama untuk proses gunakan huruf besar.
  - c. Nama untuk *data store*/simpanan data gunakan huruf besar.
  - d. Nama untuk *data flow*/arus data dengan huruf kecil

- Menjaga konsistensi dengan model lain seperti *state of purpose*, *entity relationship diagram*, *data dictionary* dan *process specification* [5].

Kesalahan dalam merancang DFD:

- Proses mempunyai *input* tetapi tidak menghasilkan *output*.  
Kesalahan ini disebut dengan *black hole* (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam.
- Proses menghasilkan *output* tetapi tidak pernah menerima *input*.  
Kesalahan ini disebut dengan *miracle* (ajaib), karena secara ajaib dihasilkan *output* tanpa pernah menerima *input* [13].

#### 2.4.3 PIECES (Performances, Information, Economy, Control, Efficiency, Dan Services)

PIECES digunakan untuk mengidentifikasi masalah, analisis terhadap kinerja informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Dengan menggunakan analisis PIECES akan mempermudah dalam menentukan kelebihan dan kekurangan yang terdapat pada sistem, sehingga dapat mempermudah untuk memperbaiki kesalahan pada sistem [14]. Metode analisis PIECES sebagai berikut [14]:

- Analisis Kinerja (*Performances*), *performance* diukur dengan jumlah produksi (*throughput*) dan waktu tanggap (*respon time*) dari suatu sistem. Jumlah produksi adalah jumlah pekerjaan yang bisa diselesaikan selama jangka waktu tertentu. Waktu tanggap adalah keterlambatan rata-rata antara suatu transaksi dengan tanggapan yang diberikan kepada transaksi tersebut.
- Analisis Informasi (*Information*), evaluasi terhadap kemampuan sistem informasi dalam menghasilkan informasi yang bermanfaat. Analisis informasi diukur dengan keakuratan dan tepat waktunya dalam menangani masalah yang muncul.
- Analisis Ekonomi (*Economy*), menganalisis dari segi biaya yang keluar maupun potensi manfaat/profit yang mungkin diperoleh. Persoalan ekonomis dan peluang berkaitan dengan masalah biaya.

4. Analisis Keamanan (*Control*), kontrol dipasang untuk meningkatkan kinerja sistem, mencegah atau mendeteksi kesalahan sistem, menjamin kesalahan data, informasi, dan persyaratan.
5. Analisis Efisiensi (*eficiency*), efisiensi merupakan cara bagaimana menghasilkan *output* sebanyak-banyaknya dengan *input* yang sekecil mungkin.
6. Analisis Layanan (*Services*), beberapa penilaian dimana kualitas suatu sistem bisa dikatakan buruk, yaitu:
  - a. Sistem menghasilkan produk yang tidak akurat.
  - b. Sistem menghasilkan produk yang tidak konsisten.
  - c. Sistem menghasilkan produk yang tidak dipercaya.
  - d. Sistem tidak mudah dipelajari.
  - e. Sistem tidak mudah digunakan.
  - f. Sistem canggung untuk digunakan.
  - g. Sistem tidak fleksibel.

#### 2.4.4 Kamus Data

Kamus data adalah kumpulan *file* yang berisi metadata *database*. Kamus data berisi catatan tentang objek basis data, seperti kepemilikan data, hubungan data dengan objek lain dan data lainnya. Dengan menggunakan kamus data ini maka suatu *Database Management Systems* (DBMS) menyimpan data secara konsisten sehingga mengurangi data ganda. Program aplikasi tidak harus merinci karakteristik data yang diperlukan dari basis data, hal ini memungkinkan bagi pemakai untuk mengubah karakteristik dari unsur data dalam kamus data tanpa harus mengubah seluruh program aplikasi yang menggunakan unsur data tersebut. Selain digunakan untuk dokumentasi dan mengurangi redundansi dalam basis data, kamus data juga dapat digunakan untuk [1]:

1. Validasi diagram aliran data untuk kelengkapan dan akurasi.
2. Memberikan titik awal untuk mengembangkan layar dan laporan.
3. Menentukan isi data yang disimpan dalam *file*.
4. Mengembangkan logika untuk proses diagram aliran data.
5. Buat bahasa markup yang dapat diperluas (XML).

Kamus data dapat diklasifikasi menjadi aktif atau pasif. Kamus data yang aktif diperbarui secara otomatis oleh DBMS (*database management system*) dengan setiap akses basis data untuk menjaga agar informasi aksesnya *terupdate*. Kamus data pasif tidak diperbarui secara otomatis dan biasanya memerlukan proses *batch* yang berjalan. Informasi akses kamus data biasanya digunakan oleh DBMS untuk optimasi *query*. Fungsi utama kamus data adalah untuk menyimpan deskripsi dari semua objek yang berinteraksi dengan *database*. Kamus data terintegrasi cenderung membatasi *metadata* pada data yang dikelola oleh DBMS. Sistem kamus data yang berdiri sendiri biasanya lebih fleksibel dan memungkinkan DBA (*database administrators*) untuk mendeskripsikan dan mengelola semua data organisasi, baik yang terkomputerisasi atau tidak. Apa pun format kamus data, ia menyediakan perancang basis data dan pengguna akhir dengan kemampuan yang jauh lebih baik untuk berkomunikasi. Selain itu, kamus data adalah alat yang membantu DBA menyelesaikan konflik data [15]. Meskipun tidak ada format standar untuk informasi yang disimpan dalam kamus data, beberapa fitur umum digunakan. Misalnya, kamus data biasanya menyimpan deskripsi berikut:

1. Elemen data yang didefinisikan di semua tabel dari semua *database*. Secara khusus, kamus data menyimpan nama elemen, tipe data, format tampilan, format penyimpanan internal, dan aturan validasi. Kamus data menjelaskan di mana suatu elemen digunakan, siapa yang menggunakannya, dan sebagainya.
2. Tabel didefinisikan di semua *database*. Misalnya, kamus data cenderung menyimpan nama pembuat tabel, tanggal pembuatan, hak akses, dan jumlah kolom.
3. Indeks yang ditentukan untuk setiap tabel *database*. Untuk setiap indeks, DBMS menyimpan setidaknya nama indeks, atribut yang digunakan, lokasi, karakteristik indeks tertentu, dan tanggal pembuatan.
4. *Database* yang ditentukan. Informasi ini mencakup siapa yang membuat setiap *database*, kapan *database* dibuat, dimana *database* berada, nama DBA, dan seterusnya.
5. Pengguna akhir dan *administrator database*. Informasi ini mendefinisikan pengguna *database*.

6. Program yang mengakses *database*. Informasi ini mencakup format layar, format laporan, program aplikasi, dan kueri SQL.
7. Hak akses untuk semua pengguna dari semua *database*. Informasi ini mendefinisikan siapa yang dapat memanipulasi objek dan jenis operasi apa yang dapat dilakukan.
8. Hubungan antar elemen data. Informasi ini mencakup elemen mana yang terlibat, apakah hubungan itu wajib atau opsional, dan persyaratan konektivitas [15].

Pendefinisian elemen-elemen data menggunakan notasi yang umumnya digunakan dalam menganalisis sistem dengan sejumlah simbol yaitu:

Tabel 2. 1 Simbol Kamus Data

No	Simbol	Uraian
1.	=	Terdiri dari, mendefinisikan, diuraikan menjadi
2.	+	Dan
3.	()	Menunjukkan suatu elemen yang bersifat pilihan ( <i>optional</i> ). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk <i>field numeric</i> pada struktur <i>file</i> .
4.	{ }	Menunjukkan elemen-elemen <i>repetitive</i> , juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau berapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, 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 secara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah atau sama lain (dengan kata lain, memiliki salah satu dari jumlah alternatif, seleksi).

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

Contoh kamus data penjualan menggunakan *website*

1. *Register* = *Username + Password + Repeat\_Password + Name+E-mail + Phone*
2. *Login* = *Username + Password*
3. *Order* = *Order Id + Order\_Date + { Id\_Menu + Menu + Qty + Price + Subtotal } + Total\_Order + Discount +Promotion\_Code*
4. *Payment\_Method* = *[Bank\_Bri|Bank\_Bca|Mybank]*
5. *Confirmation Payment* = *Order ID+Account\_Owner+ Payment\_Receipt + Transfer\_Date*

## 2.5 Normalisasi

Normalisasi adalah cara yang digunakan untuk menurunkan jumlah duplikasi data dan anomali data pada basis data sehingga basis data dapat bekerja dengan optimal. Normalisasi berperan sebagai teknik validitas untuk memeriksa apakah struktur hubungan desain basis data baik atau tidak [16]. Normalisasi digunakan untuk mengembangkan keseluruhan desain basis data yang sederhana, fleksibel, dan bebas dari redundansi data. Normalisasi melibatkan penerapan seperangkat aturan yang dapat membantu mengidentifikasi dan memperbaiki masalah dan kompleksitas yang melekat dalam desain tabel. Tujuan dilakukannya normalisasi yaitu:

1. Meminimalisasi redundansi atau perulangan data.
2. Memecah tabel ke beberapa tabel-tabel.
3. Mempercepat *update*, *insert*, dan *delete*.

Beberapa *key* dalam normalisasi:

1. *Superkey*, adalah sejumlah *attribute entity* yang dapat digunakan untuk mengidentifikasi objek secara unik.
2. *Candidate Key* adalah *superkey* dengan jumlah *attribute* minimal dan dapat berdiri sendiri.

3. *Primary Key* adalah *superkey* yang dipilih oleh desainer atau *database administrator*.
4. *Foreign Key* adalah *attribute* disuatu relasi (tabel) yang menjadi *primary key* di relasi (tabel) lain [17].

Terdapat beberapa jenis normalisasi namun sebagian besar yang melakukan perancangan basis data hanya sampai 3NF karena sudah cukup memadai dalam menghasilkan table-table yang baik. Terdapat beberapa jenis normalisasi yaitu:

1. Normalisasi 1NF (*First Normal Form*)

Bentuk normal pertama terpenuhi jika sebuah tabel tidak memiliki atribut yang bernilai banyak (*multi value attribute*) artinya setiap pertemuan baris dan kolom hanya berisikan satu nilai (*single value attribute*). Tujuan dari 1NF adalah membuang adanya redundansi data, dan menghindari adanya pencatatan *null value* [18].

BON PEMBELIAN									
No Faktur	Tanggal	No Polisi	Warna	Merek	Tahun	Mekanik ID	Nama Mekanik	Kode Parts	
05103214	25/10/2005	B3117LB	Biru	Supra X	2005	DDE	Diko Dewanto	20W501000CC	..
05103214	25/10/2005	B3117LB	Biru	Supra X	2005	DDE	Diko Dewanto	SERV001	..
05103215	25/10/2005	B2121AA	Merah	Supra X	2005	DDE	Diko Dewanto	SERV001	..

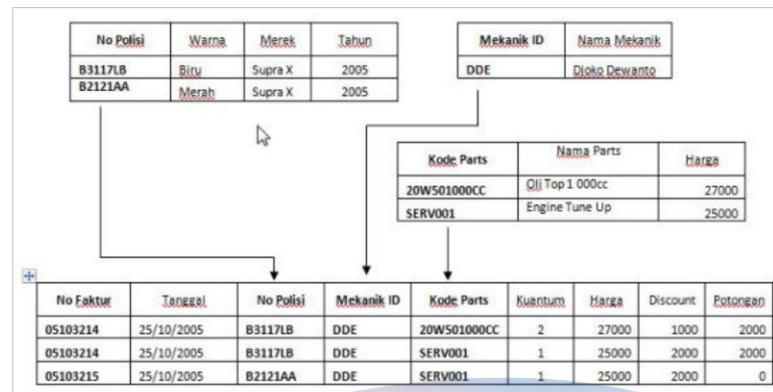
  

Nama Parts	Kuantum	Harga	Discount	Jumlah	Potongan	Total
Oli Top 1 000cc	2	27000	1000	52000	2000	73000
Engine Tune Up	1	25000	2000	23000	2000	73000
Engine Tune Up	1	25000	2000	23000	0	23000

Gambar 2. 4 Contoh Ilustrasi 1NF

2. Normalisasi 2NF (*Second Normal Form*)

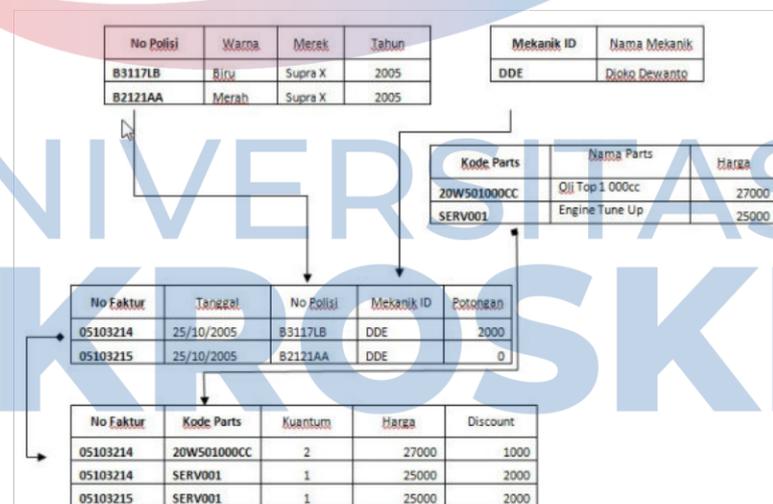
Bentuk normal kedua akan terpenuhi jika bentuk data telah memenuhi kriteria bentuk normal pertama dan setiap atribut yang bukan kunci haruslah bergantung secara fungsional (*functional dependency*) terhadap atribut kunci/*primary key*. Sehingga untuk membentuk normal ke dua haruslah sudah ditentukan *field key* [18].



Gambar 2. 5 Contoh Ilustrasi 2NF

### 3. Normalisasi 3NF (*Third Normal Form*)

Bentuk normal ketiga telah memenuhi kriteria bentuk normal pertama dan kedua serta tidak terdapat *transitive dependency* yaitu sebuah atribut yang bukan kunci selain bergantung kepada atribut kunci, juga bergantung kepada atribut bukan kunci lainnya. Sehingga setiap atribut bukan kunci haruslah bergantung hanya pada atribut kunci (*primary key*) [18].



Gambar 2. 6 Contoh Ilustrasi 3NF

## 2.6 Basis Data

Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan perangkat lunak untuk melakukan manipulasi untuk kegunaan tertentu [19]. Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik [19].

Basis data mempunyai beberapa kriteria penting, yaitu bersifat *data oriented* dan bukan *program oriented*, dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya, dapat berkembang dengan mudah baik volume maupun strukturnya, dapat memenuhi kebutuhan sistem-sistem baru secara mudah, dapat digunakan dengan cara-cara yang berbeda [19].

Tujuan dari Basis Data:

1. Mengatur data/mengorganisasikan data agar diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali.
2. Tidak ada duplikasi data sehingga konsistensi data mudah dijaga.
3. Data terintegrasi.
4. Data tidak tergantung pada program aplikasi, sehingga pemeliharaan program aplikasi mudah dilakukan.
5. Data dapat dipakai secara bersama oleh beberapa pemakai.
6. Dapat diterapkan standarisasi.
7. Informasi selalu mutakhir (*up-to-date*).
8. Untuk memenuhi kebutuhan-kebutuhan konten informasi dari pengguna dan aplikasi-aplikasi tertentu.
9. Menyediakan struktur informasi yang alami dan mudah dipahami.
10. Mendukung kebutuhan-kebutuhan pemrosesan dan objektivitas kinerja (waktu respon, waktu pemrosesan, dan ruang penyimpanan) [20].

## 2.7 Panti Asuhan

Panti asuhan merupakan suatu lembaga usaha kesejahteraan sosial pada anak yang terlantar dengan melaksanakan penyantunan dan pengentasan anak terlantar, memberikan pelayanan pengganti orang tua/wali anak dalam memenuhi kebutuhan

fisik, mental dan sosial kepada anak asuh sehingga memperoleh kesempatan yang luas, tepat dan memadai bagi pengembangan kepribadiannya sesuai dengan yang diharapkan sebagai bagian dari generasi penerus cita-cita bangsa dan sebagai insan yang akan turut serta aktif dalam bidang pembangunan nasional [21]. Peranan panti asuhan sebagai bekal hidup, yang berupa bimbingan antara lain:

1. Bimbingan Kemandirian

Anak panti asuhan dapat melaksanakan semua kegiatan sendiri tanpa harus bergantung dengan orang lain, anak panti dituntut untuk disiplin pada dirinya masing-masing dan bertanggung jawab dengan apa yang dilakukan maupun ditugaskan.

2. Memberikan Bekal Hidup

Anak panti asuhan dibekali dengan pendidikan dan ilmu agama sehingga anak-anak di panti asuhan memiliki kehidupan yang lebih baik, memiliki nilai-nilai dan etika yang baik sehingga bisa mencegah anak-anak panti asuhan dari tindakan kriminalitas.

3. Memberikan Pelayanan Sosial kepada Anak-Anak Asuh Panti

Tempat yang paling sesuai untuk untuk menerima anak yatim dan terlantar. Tempat ini juga dapat memenuhi kebutuhan sehari-hari dan kebutuhan pendidikan.

4. Memberikan Bimbingan Fisik dan Mental

Melakukan senam di pagi hari untuk menjaga kesehatan anak-anak panti asuhan dan memberikan pendidikan agama berupa tadarusan, akhlaq dan budi pekerti, akidah dan lain sebagainya [22].

### 2.7.1 Donatur

Donatur merupakan orang yang memberikan donasi berupa uang secara tetap kepada suatu perkumpulan dan sebagainya. Donatur juga berupa kelompok maupun lembaga yang memiliki ketertarikan dan kesediaan untuk memberikan donasi berupa jasa, barang, dan pendanaan [23]. Beberapa tipe donatur seperti:

1. Donatur Tetap

Donatur yang memberikan sumbangan secara rutin. Dimana donatur ini memiliki komitmen dalam memberikan sumbangan, misalnya donatur

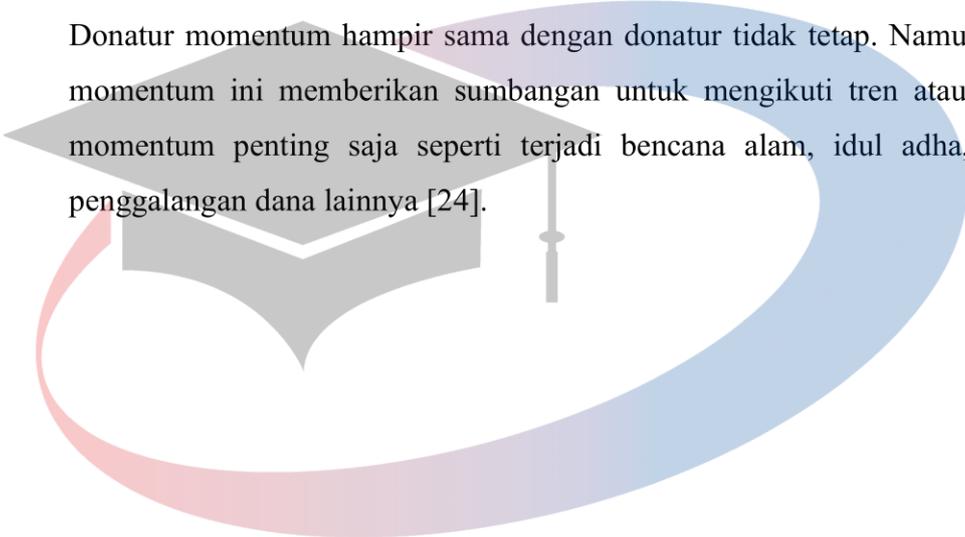
berkomitmen agar memberikan sumbangan tiap bulan dengan jumlah Rp 100.000.

2. Donatur Tidak Tetap/*Insidental*

Donatur yang jumlah dan waktu donasinya tidak ditentukan, misalnya dalam 1 bulan donatur hanya menyumbangkan Rp10.000 kemudian donatur menyumbang lagi 2 bulan setelahnya dengan nominal Rp5.000.

3. Donatur Momentum

Donatur momentum hampir sama dengan donatur tidak tetap. Namun donatur momentum ini memberikan sumbangan untuk mengikuti tren atau saat ada momentum penting saja seperti terjadi bencana alam, idul adha, maupun penggalangan dana lainnya [24].



UNIVERSITAS  
MIKROSKIL