

## BAB II TINJAUAN PUSTAKA

### 2.1 Konsep Sistem Informasi

#### 2.1.1 Sistem

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama – sama. Sistem terbentuk oleh subsistem-subsistem yang saling terhubung dan mendukung untuk mencapai sasaran dan menghasilkan sebuah hasil (*output*). Model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu, sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem [1]. Adapun karakteristik yang dimaksud adalah sebagai berikut [2]:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan serupa supra sistem.

b. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya, batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan, karena kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem(*interface*)

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian, terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan sistem(*input*)

Masukan adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer program.

f. Keluaran Sistem(*Output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi sistem yang lain.

g. Pengolah Sistem (*Process*)

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

h. Sasaran Sistem(*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministic. Kalau sistem yang tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

### 2.1.2 Informasi

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat. Proses pengolahan ini memerlukan teknologi. Pada proses pengolahan data, untuk dapat menghasilkan informasi, juga dilakukan verifikasi secara akurat, spesifik, dan tepat waktu [1]. Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Kualitas dari suatu informasi tergantung dari 3 (tiga) hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*) [2].

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak menyesatkan. Akurat berarti juga informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi ini sampai ke penerima informasi bisa saja mengalami banyak gangguan yang dapat mengubah atau merusak informasi.

## 2. Tepat Waktu (*timelines*)

Informasi yang sampai kepada si penerima tidak boleh terlambat. Informasi yang sudah sangat usang tidak akan mempunyai nilai karena informasi merupakan landasan dalam pengambilan keputusan. Apabila pengambilan keputusan terlambat maka dapat berdampak fatal bagi perusahaan. Sekarang ini informasi bernilai mahal karena harus cepat dikirim dan didapatkan sehingga memerlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkannya.

## 3. Relevan (*relevance*)

Informasi mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda. Menyampaikan informasi tentang kerusakan mesin produksi kepada akuntan perusahaan tentunya kurang relevan. Akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi disampaikan kepada ahli teknik perusahaan informasinya kurang relevan, tetapi akan sangat relevan untuk akuntan perusahaan.

### 2.1.3. Sistem Informasi

Sistem Informasi merupakan suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi sebagai suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan.

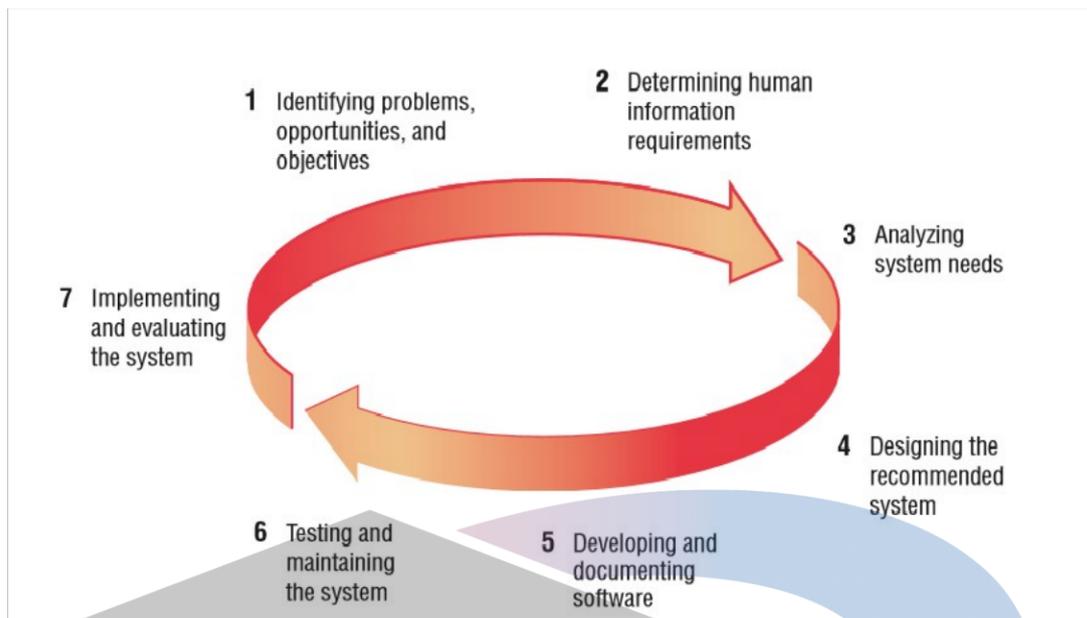
Sistem informasi menerima masukan data, instruksi dan mengolah data sesuai dengan perintah untuk mengeluarkan hasilnya, ini merupakan sebagian dari peristiwa yang terjadi pada sistem informasi. Didalam Sistem Informasi juga terdapat proses perencanaan, control, koordinasi, dan pengambilan keputusan. Sehingga sebagai sebuah sistem yang mengelolah data menjadi informasi yang akan disajikan dan digunakan oleh pengguna, maka sistem informasi merupakan sebuah sistem yang kompleks [3].

Sistem informasi memiliki fungsi utama antara lain:

1. Sistem informasi sama pentingnya dengan fungsi-fungsi utama dalam bisnis seperti akuntansi, keuangan, manajemen operasional, pemasaran dan manajemen sumber daya manusia.
2. Sistem informasi berperan sebagai kontribusi bagi efisiensi operasional perusahaan, produktivitas dan kinerja karyawan, serta layanan dan kepuasan pelanggan.
3. Sistem informasi menjadi sumber nilai bagi perusahaan.
4. Sistem informasi dibutuhkan untuk mendukung pembuatan keputusan yang efektif oleh manajer dan pelaku bisnis.
5. Sistem informasi merupakan unsur penting dalam mengembangkan produk dan layanan kompetitif yang memberikan keunggulan strategis bagi perusahaan dalam pasar global.
6. Sistem informasi memberikan kesempatan karier yang terbuka, prestisius, dan menantang bagi banyak orang.
7. Sistem informasi merupakan komponen penting dalam hal sumber daya, infrastruktur, dan kemampuan perusahaan bisnis berjejaring di masa kini.
8. Sistem informasi merupakan sumber daya yang strategis [4].

## 2.2 SDLC (System Development Life Cycle)

SDLC (*System Development Life Cycle*) atau disebut juga dengan siklus hidup pengembangan sistem adalah pendekatan bertahap untuk analisis dan perancangan yang menyatakan sistem paling baik dikembangkan melalui penggunaan siklus khusus kegiatan analisis dan pengguna. Analisis tidak setuju dengan persis berapa banyak fase yang ada di SDLC, tetapi mereka umumnya memuji pendekatan terorganisir. Disini 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 [5].



Gambar 2. 1 *System Development Life Cycle (SDLC)*

Tahapan-tahapan dalam siklus hidup pengembangan terdiri dari:

1. Mengidentifikasi masalah, peluang dan tujuan

Dalam fase pertama siklus hidup pengembangan sistem ini, analisis berkepentingan untuk mengidentifikasi dengan benar masalah, peluang dan tujuan. Tahap ini penting bagi keberhasilan proyek, karena tidak ada yang mau menyia-nyiakan waktu berikutnya untuk mengatasi masalah yang salah.

Fase pertama mengharuskan analisis untuk melihat dengan jujur apa yang terjadi dalam bisnis. Kemudian, bersama dengan anggota organisasi lainnya, analisis menunjukkan masalah. Sering kali orang lain akan memunculkan masalah ini, dan mereka adalah alasan analisis awalnya dipanggil. Peluang adalah situasi yang diyakini analisis 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 analisis 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, analisis, 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 grub pengguna tidak memiliki dana yang cukup dalam anggarannya atau ingin mengatasi masalah yang tidak terkait, atau jika masalah tidak memerlukan sistem computer, solusi yang berbeda mungkin disarankan, dan proyek sistem tidak melanjutkan 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 pengambilan keputusan dan lingkungan kantor mereka, dan metode yang mencakup semua, seperti *prototyping*.

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 system 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 membantu sistem yang bermanfaat bagi pekerjaan 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 (aktifitas bisnis), dan dimana (lingkungan dimana pekerjaan berlangsung), kapan (waktu), dan bagaimana (bagaimana prosedur saat ini dilakukan) dari bisnis yang diteliti. Analisis kemudian harus bertanya mengapa bisnis menggunakan system saat ini. Mungkin ada alasan yang bagus untuk melakukan bisnis menggunakan metode saat ini, dan ini harus dipertimbangkan ketika merancang sistem baru.

## 3. Menganalisis kebutuhan-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, proses, dan output fungsi bisnis, atau diagram aktifitas atau diagram urutan untuk menunjukkan urutan peristiwa, menggambarkan system 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, analisis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk mencapai desain logis dari sistem informasi. Analisis merancang prosedur bagi pengguna untuk membantu mereka memasukkan data secara akurat sehingga data yang masuk ke sistem informasi benar. Selain itu, analisis menyediakan bagi pengguna untuk menyelesaikan input efektif ke sistem informasi dengan menggunakan Teknik bentuk yang baik dan halaman web 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 analisis juga bekerja dengan pengguna untuk mendasi output (baik pada layar atau dicetak) yang memenuhi kebutuhan informasi mereka.

Akhirnya, analisis harus merancang control dan prosedur cadangan untuk melindungi sistem dan data, 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, nama dan fungsi dari setiap kode yang akan ditulis sebelumnya baik yang ditulis sendiri atau menggunakan kode atau Pustaka kelas lainnya.

#### 5. Mengembangkan dan mendokumentasikan perangkat lunak

Pada fase kelima SDLC, analisis bekerja dengan programmer untuk mengembangkan perangkat lunak asli yang diperlukan. Selama fase ini analisis bekerja dengan pengguna untuk mengembangkan dokumentasi yang efektif untuk perangkat lunak, termasuk manual prosedur, bantuan online, dan situs web yang menampilkan *Frequently Asked Questions* (FAQ), pada file *read me* yang akan dikirimkan dengan perangkat lunak baru. Karena pengguna terlibat sejak awal, dokumentasi fase harus menjawab pertanyaan yang mereka ajukan dan selesaikan bersama dengan analisis. Dokumentasi memberi tahu pengguna cara menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi masalah perangkat lunak.

Programmer mencukur peran kunci dalam fase ini karena mereka merancang, kode, dan menghapus kesalahan sintaksis 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 masuk ke pengguna. Beberapa pengujian diselesaikan oleh pemrograman saja, sebagian lagi oleh analisis sistem bersama dengan pemrogram. Pemeriksaan tes untuk menunjukkan masalah dijalankan

pertama dengan data sampel dan akhirnya dengan data actual dari sistem saat ini. Seringkali rencana pengujian dibuat di awal SDLC dan disempurnakan seiring proyek berlangsung.

Pemeliharaan diselesaikan oleh pemrogram saja, sebagian lagi oleh analisis sistem bersama dengan pemrogram

#### 7. Mengimplementasikan dan mengevaluasi sistem

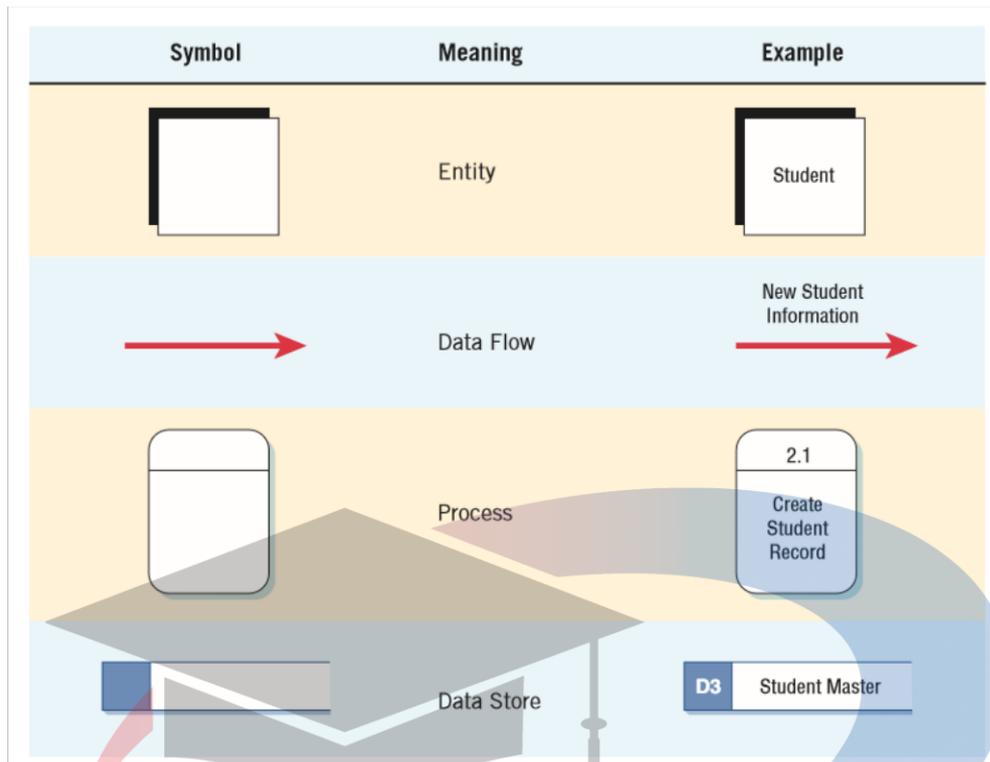
Dalam fase terakhir pengembangan sistem ini, analisis membantu mengimplementasikan sistem informasi. Fase ini melibatkan melatih pengguna untuk menangani sistem. Vendor melakukan beberapa pelatihan, tetapi pengawasan pelatihan adalah tanggung jawab analisis sistem. Selain itu, analisis perlu merencanakan konversi yang lancar dari sistem lama ke yang baru. Proses ini termasuk mengkonversi file dari format lama ke yang baru, atau membangun database, memasang peralatan, dan membawa sistem baru kedalam produksi.

Evaluasi dimasukkan sebagai bagian dari fase akhir SDLC ini Sebagian besar demi diskusi. Sebenarnya, evaluasi berlangsung selama setiap fase. Kriteria utama yang harus dipenuhi adalah apakah pengguna yang dituju memang menggunakan sistem. Perlu dicatat bahwa kerja sistem sering kali bersifat siklis. Ketika seorang analisis menyelesaikan satu fase pengembangan sistem dan melanjutkan ke tahap berikutnya, penemuan masalah dapat memaksa analisis untuk Kembali ke fase sebelumnya dan memodifikasi pekerjaan yang dilakukan di sana [5].

## 2.3 Teknik Pengembangan Sistem

### 2.3.1 Data Flow Diagram (DFD)

*Data flow diagram* atau disebut juga diagram aliran data adalah suatu diagram yang menggambarkan pandangan mengenai ,masukan, proses, dan keluaran sistem. Penganalisis sistem berupaya memahami syarat-syarat informasi penggunaan, yaitu harus mampu mengonseptualisasikan bagaimana data-data berpindah didalam perusahaan, proses-proses atau transformasi dimana data-data keluarannya. Dengan menggunakan kombinasi dari empat simbol, penganalisis sistem dapat menciptakan suatu gambaran prose – proses yang bisa menampilkan dokumentasi sistem solid [5].



**Gambar 2. 2** Simbol Pada Data Flow Diagram

Keempat simbol dasar yang digunakan untuk menetapkan gerakan diagram aliran data adalah [5]:

### 1. Entity

Digunakan untuk menggambarkan suatu entitas eksternal yang dapat mengirimkan data atau menerima data dari sistem. Entitas eksternal atau hanya entitas disebut juga sumber atau tujuan data dan dianggap eksternal terhadap sistem yang sedang digambarkan. Setiap entitas diberi label dengan sebuah nama yang sesuai. Entitas yang sama bisa digunakan lebih dari sekali atas suatu diagram aliran data tertentu hal ini agar menghindari persilangan antara jalur – jalur aliran data.

### 2. Data Flow

Untuk menunjukkan perpindahan data dari titik ke titik yang lainnya, dengan kepala tanda panah mengarah ke tujuan. Karena tanda panah menunjukkan seseorang, tempat, atau sesuatu, maka harus digambarkan dalam kata benda.

### 3. Process

Untuk menunjukkan adanya proses transformasi. Proses – proses tersebut selalu menunjukkan suatu perubahan jadi aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk.

#### 4. Data Store

Bujur sangkar dengan ujung tertutup pada sisi sebelah kiri dan terbuka pada sisi sebelah kanan untuk menunjukkan data *store* (tempat penyimpanan data), dapat berupa *file* atau dokumen.

Kelebihan pendekatan diagram aliran data, yaitu:

1. Kebebasan dari menjalankan implementasi teknis sistem yang dini.
2. Pemahaman lebih jauh mengenai ketertarikan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data
4. Menganalisis sistem yang diajukan untuk menentukan apakah data dan proses yang diperlukan.

Kesalahan didalam penggambaran diagram aliran data yaitu:

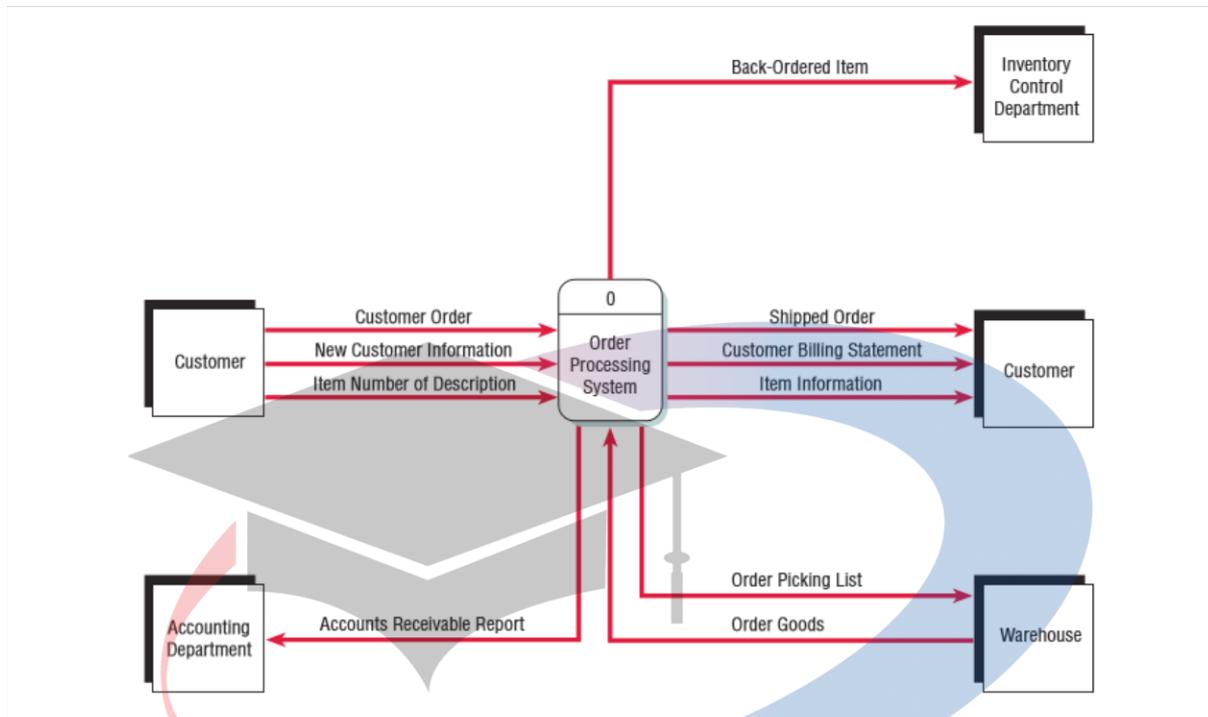
1. Lupa memasukkan suatu aliran data atau mengarahkan anak panah kearah yang salah. Setiap proses mentransformasikan data dan harus menerima masukan dan menghasilkan keluaran.
2. Menghubungkan penyimpanan data dan entitas eksternal secara langsung tanpa melalui suatu proses.
3. Pemberian label pada aliran data atau proses yang tidak tepat.
4. Memasukkan lebih dari Sembilan proses pada diagram aliran data sehingga menghasilkan suatu diagram yang kacau. Hal ini akan memusingkan pembaca dan menghalangi komunikasi.
5. Menghilangkan aliran data. Periksa apakah diagram terdapat aliran *linear*, artinya suatu proses yang hanya mempunyai satu masukan dan satu keluaran.
6. Dekomposisi yang tidak seimbang pada diagram anak. Masing-masing diagram anak harus memiliki masukan dan keluaran yang sama seperti proses induk.

Langkah-langkah dalam mengembangkan Diagram Aliran Data sebagai berikut [5]:

#### 1. Diagram Konteks

Dengan pendekatan atas-bawah untuk membuat diagram pengalihan data, diagram berganti dari umum ke khusus. Meskipun diagram ini membantu penganalisis sistem memahami pengalihan data, sifat umum membatasi kegunaannya. Diagram konteks awal harus berupa suatu pandangan yang mencakup masukan-masukan dasar, sistem umum dan keluaran. Diagram konteks merupakan tingkatan tertinggi dalam diagram

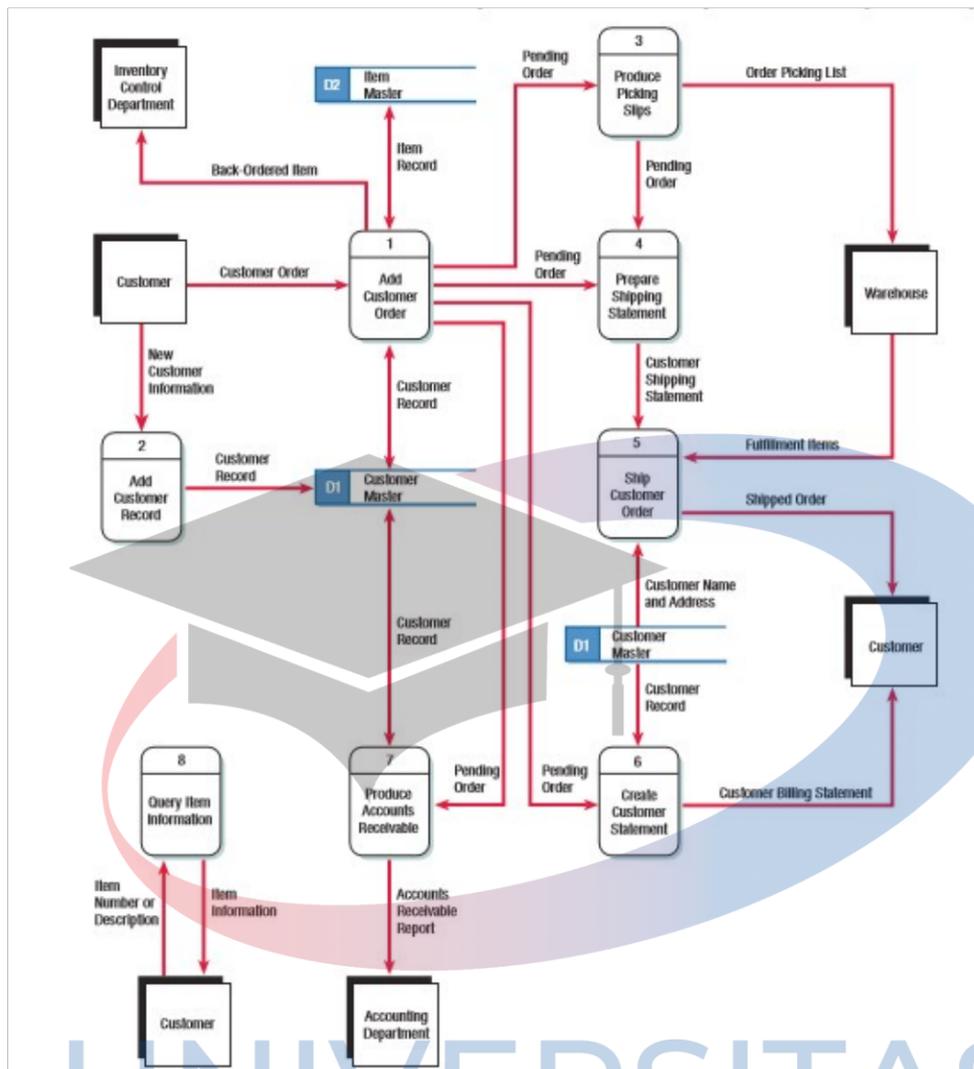
aliran data dan hanya membuat suatu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol.



**Gambar 2. 3** Contoh DFD Level Konteks

## 2. Diagram Level 0

Masukan dan keluaran yang ditetapkan dalam diagram konteks harus tetap sama dengan masukan dan keluaran di diagram nol. Sisa diagram konteks dikembangkan kedalam gambar terperinci yang melibatkan tiga sampai sembilan proses dan menunjukkan penyimpanan data dan aliran data pada level yang lebih rendah. Diagram nol merupakan pengembangan sistem diagram konteks dan bisa mencakup sampai Sembilan proses. Memasukkan lebih banyak proses pada level ini akan membuat diagram menjadi sulit dipahami. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sebelah atas diagram dan mengarah ke sudut kanan bawah.

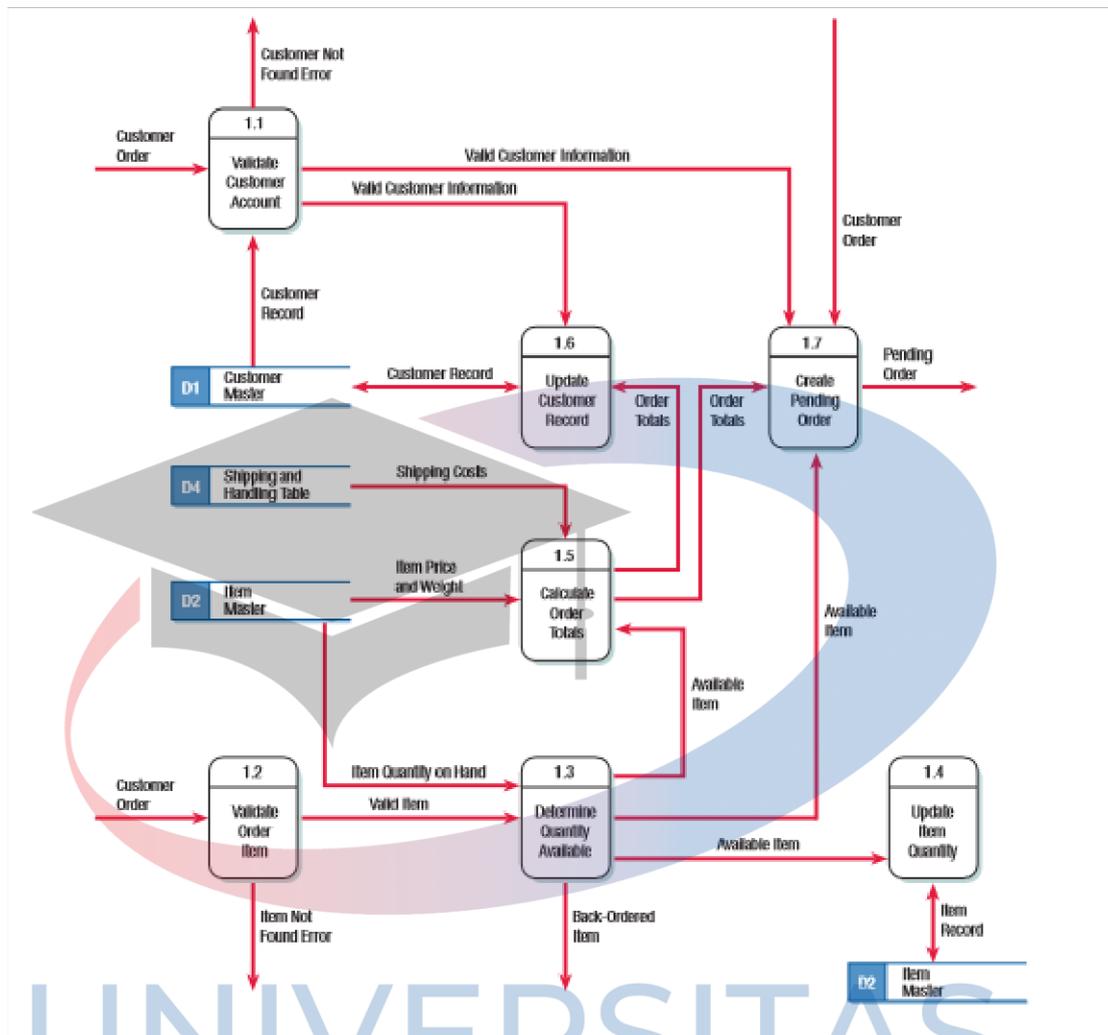


Gambar 2. 4 Contoh DFD Level 0

### 3. Diagram Level 1

Setelah selesai membuat DFD level 0, maka tahap selanjutnya adalah merinci setiap proses yang ada pada DFD level 0, sehingga *event* yang ada dalam suatu proses dapat digambarkan menjadi lebih detail dalam sebuah DFD lagi, yang disebut dengan DFD level 1. DFD level 1 bertujuan untuk memberikan pandangan mengenai keseluruhan sistem dengan lebih mendalam. Proses utama yang akan dipecahkan menjadi sub-proses, *data store* yang digunakan dalam proses-proses utama juga diidentifikasi dalam DFD level 1. Dalam membuat DFD level 1 hubungan sistemnya dan lingkungannya tidak boleh dihilangkan. Dengan kata lain arus data yang masuk ke sistem dan arus data yang keluar dari sistem harus persis sama dengan yang ada pada DFD level 0. Jika misalnya menemukan aliran data baru

ketika membuat DFD level 1 maka dalam DFD level 0 harus ditambahkan juga arus data yang baru dibuat dalam DFD level 1.



Gambar 2. 5 Contoh DFD Level 1

### 2.3.2 Fishbone Diagram

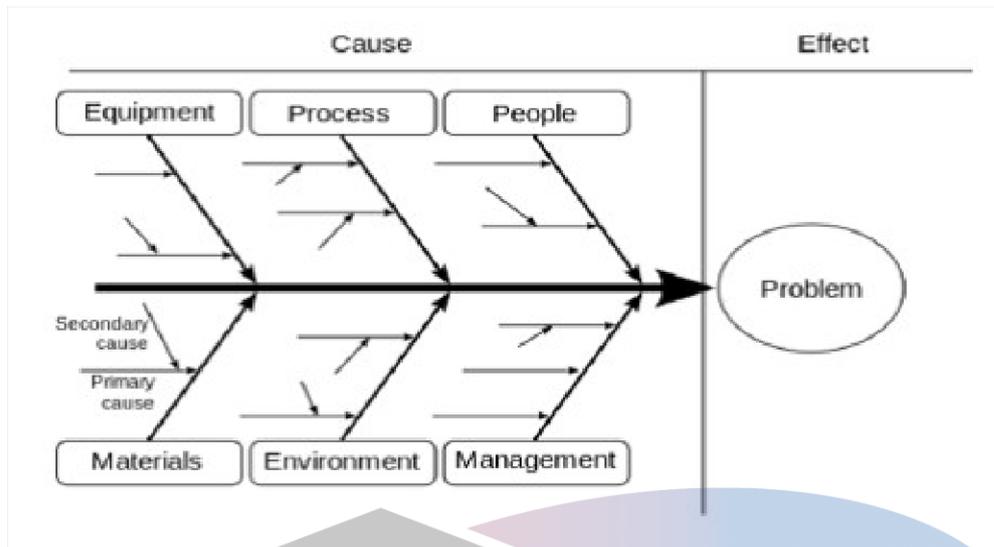
*Fishbone Diagrams* (Diagram Tulang Ikan) adalah analisis sebab akibat yang dikembangkan oleh Dr. Kaoru Ishikawa pada sekitar tahun 1960an yang menggambarkan permasalahan dan penyebabnya dalam suatu kerangka tulang ikan.

Diagram *fishbone* merupakan suatu alat visual untuk mengidentifikasi, mengeksplorasi, dan secara grafik menggambarkan secara detail semua penyebab yang berhubungan dengan suatu permasalahan. Menurut Scarvada konsep dasar dari diagram *fishbone* adalah permasalahan mendasar diletakkan pada bagian kanan dari diagram atau pada bagian kepala dari kerangka tulang ikannya. Penyebab permasalahan diletakkan pada sirip dan durinya. Kategori penyebab permasalahan yang sering digunakan sebagai mulai awalnya meliputi *materials* (bahan baku), *mechines and requipment* (mesin dan peralatan), *manpower* (sumber

daya manusia) , *methods* (metode), *enviromnet* (lingkungan), dan *measurement* (pengukuran). Keenam penyebab munculnya masalah ini sering disingkat dengan 6M. Penyebab lain dari masalah selain 6M tersebut dapat dipilih jika diperlukan. Untuk mencari penyebab dari permasalahan baik yang berasal dari 6M seperti yang dijelaskan diatas maupun penyebab yang lainnya mungkin dapat digunakan teknik *brainstorming* [6].

Diagram *Fishbone* dapat digunakan untuk menganalisis permasalahan baik pada level individu, kelompok atau perusahaan/organisasi. Terdapat banyak manfaat dari pemakaian Diagram *Fishbone* ini dalam analisis masalah. Manfaat penggunaan diagram *fishbone* antara lain [5]:

1. Memfokuskan individu, tim, atau perusahaan/organisasi untuk menganalisis permasalahan akan membantu anggota tim dalam memfokuskan permasalahan pada masalah prioritas.
2. Memudahkan dalam mengilustrasikan gambaran singkat permasalahan tim/organisasi. Diagram *fishbone* dapat mengilustrasikan permasalahan utama secara singkat sehingga tim akan mudah menangkap permasalahan utama.
3. Menentukan kesepakatan mengenai penyebab suatu masalah. Dengan menggunakan teknik *brainstorming* para anggota tim akan memberikan sumbangan saran mengenai penyebab munculnya masalah. Berbagai sumbangan saran ini akan didiskusikan untuk menentukan mana dari penyebab tersebut yang berhubungan dengan masalah utama termasuk menentukan penyebab yang dominan.
4. Membangun dukungan anggota tim untuk menghasilkan solusi. Setelah ditentukan penyebab dari masalah maka langkah untuk menghasilkan solusi akan lebih mudah mendapatkan dukungan dari anggota tim.
5. Memfokuskan tim pada penyebab masalah. Diagram *fishbone* akan memudahkan anggota tim pada penyebab masalah. Juga dapat dikembangkan lebih lanjut dari setiap masalah yang sudah ditentukan
6. Memudahkan visualisasi hubungan antara penyebab dengan masalah. Hubungan ini akan terlihat dengan mudah pada diagram *fishbone* yang dibuat.
7. Memudahkan tim beserta anggota tim untuk melakukan diskusi dan menjadikan diskusi lebih terarah pada masalah dan penyebabnya.



Gambar 2. 6 contoh *Fishbone Diagrams*

### 2.3.3 Kamus Data

Kamus data merupakan metode utama untuk menganalisis aliran data dan penyimpanan data dari sistem orientasi data. Kamus data adalah referensi data pekerjaan tentang data (*metadata*). Kamus data mengoleksi, mengordinasi dan mengkonfirmasi tentang apa bentuk data spesifik yang bertujuan untuk orang-orang yang berbeda di organisasi [5].

Pada tahap analisis, kamus data digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang ada yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakain sistem. Pada tahap perancangan sistem kamus data digunakan untuk merancang input, merancang laporan-laporan *database* [5].

Fungsi dari kamus data adalah sebagai katalog yang menjelaskan lebih detail tentang DAD yang mencakup proses, *data flow*, dan *data store*. Selain itu juga untuk menghindari penggunaan kata yang sama hal ini karena kamus data disusun berdasarkan abjad [5]

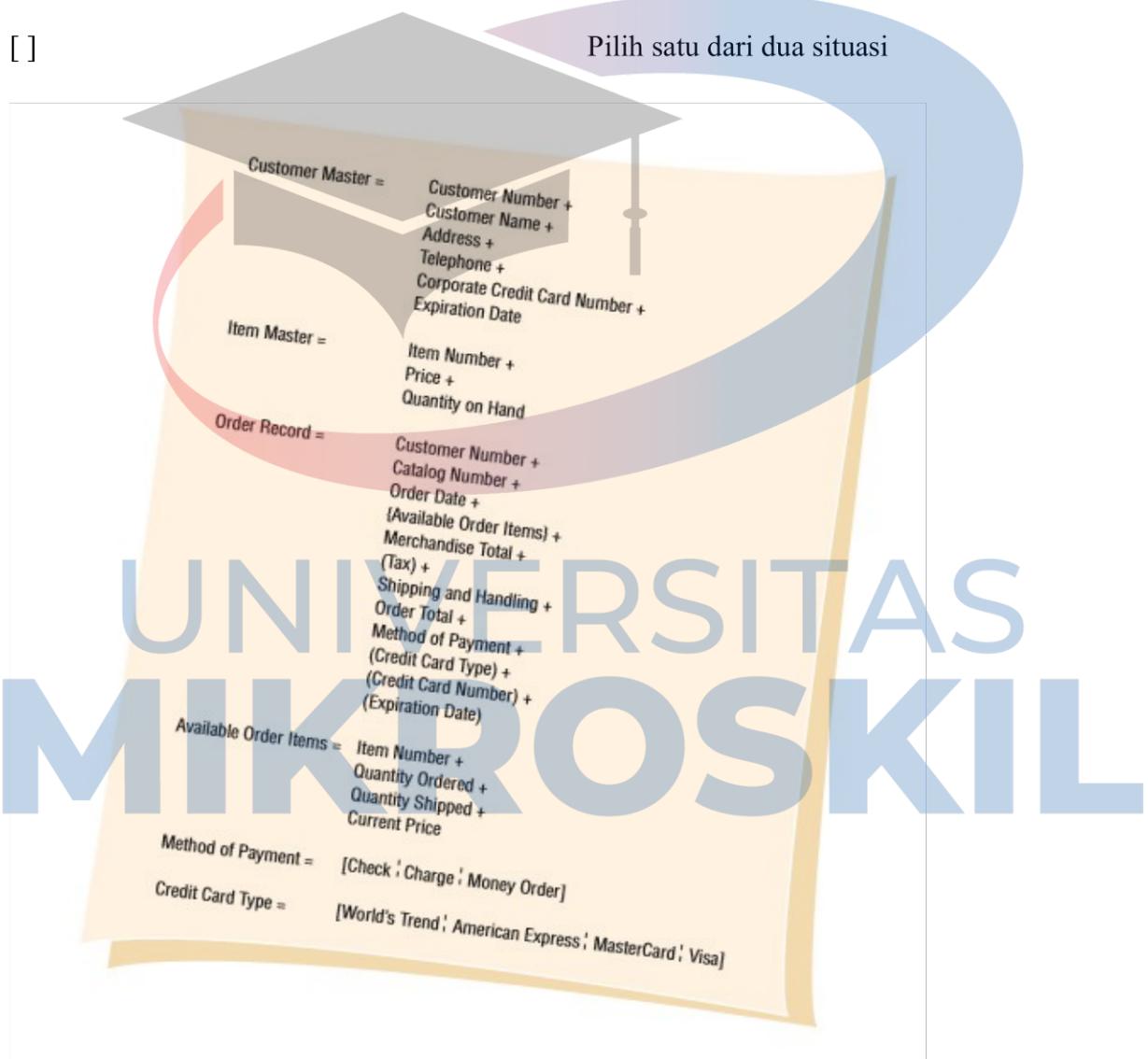
Kamus data bisa digunakan untuk [5]:

- a. Menvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
- b. Menyediakan suau titik awal untuk mengembangkan layar dan laporan-laporan
- c. Menentukan muatan data yang disimpan dalam *file-file*.
- d. Mengembangkan logika untuk proses-proses diagram aliran data.
- e. Mengurangi redudensi
- f. Dokumentasi

Menggambarkan struktur data menggunakan notasi aljabar dengan simbol-simbol sebagai berikut [5]:

Tabel 2. 1 Simbol Pada Kamus Data

Notasi	Keterangan
=	Terdiri dari
+	Dan
()	Pilihan Optional
{ }	Iterasi (Peluang Proses)
[ ]	Pilih satu dari dua situasi



Gambar 2. 7 Contoh Penulisan Kamus Data

### 2.3.4 Basis Data

Basis data biasa disebut dengan DBMS adalah kumpulan alat, fitur, dan antarmuka yang memungkinkan pengguna untuk menambah, memperbarui, mengelola, mengakses, dan menganalisis konten dari set data. Dari sudut pandang pengguna, keuntungan utama dari DBMS adalah ia menawarkan akses data yang tepat waktu, interaktif, dan fleksibel [7].

DBMS menyediakan antarmuka antara database dan pengguna yang perlu mengakses data. Meskipun pengguna terutama memperhatikan antarmuka yang mudah digunakan dan dukungan untuk kebutuhan bisnis mereka, seorang analis sistem harus memahami semua komponen DBMS. Selain antarmuka untuk pengguna, administrator basis data, dan sistem terkait, DBMS juga memiliki bahasa manipulasi data, skema, dan sub-skema, dan *repository* data fisik. Tujuan efektivitas *database* meliputi:

1. Memastikan bahwa data dapat dibagikan diantara pengguna untuk berbagai aplikasi
2. Menjaga data yang akurat dan konsisten
3. Memastikan bahwa semua data yang diperlukan untuk aplikasi saat ini dan masa depan akan tersedia.
4. Mengizinkan *database* untuk berkembang seiring dengan meningkatkan kebutuhan pengguna.
5. Mengizinkan pengguna untuk membangun pandangan pribadi mereka tentang data tanpa memperhatikan cara data disimpan secara fisik.

Pendekatan basis data memiliki keuntungan yaitu memungkinkan pengguna untuk memiliki pandangan mereka sendiri tentang data. Pengguna tidak perlu khawatir dengan struktur aktual dari *database* pusat dari *mainframe* dan mengunduhnya ke *PC* atau perangkatn genggam. Basis data yang lebih kecil ini kemudian digunakan untuk menghasilkan laporan atau menjawab pertanyaan untuk pengguna akhir [7].

Urutan komponen pembentu basis data adalah sebagai berikut [8]:

1. *Characters*, bagian terkecil dapat berupa karakter menarik, huruf maupun karakter khusus yang membantu suatu item data/*filed*
2. *Field*, mempresentasikan suatu atribut dari suatu *record* yang menunjukkan suatu item dari data
3. *Record*, menggambarkan suatu unit data individu yang tertentu
4. *File*, terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis.
5. *Database*, kumpulan dari berbagai macam tipe *record* yang mempunyai hubungan terhadap suatu objek tertentu.

6. *Database system*, merupakan sekumpulan basis data yang tersusun dari beberapa file.
7. *Key*, kunci sebagai penghubung dengan tabel lain dan kunci dapat digunakan untuk membedakan relasi yang terjadi antara data pada suatu basis data. Terdapat tiga *key* didalam sebuah basis data:
  - a. *Primary key* (kunci primer)  
suatu *field* yang hanya mengidentifikasi suatu nilai dan hanya satu *record* dalam sebuah *file*. Dalam suatu prohram sering disebut dengan kunci unik (*unique key*).
  - b. *Secondary key* (kunci sekunder)  
Kunci yang mengidentifikasikan kunci alternative pada suatu basis data dan akan digunakan jika kunci primer tidak berfungsi.
  - c. *Foreign key* (kunci asing)  
Penunjukan *record* yang terdapat pada *file* yang berbeda pada suatu basis data. Penggunaan kunci asing ini dimaksudkan sebagai media penghubung *record* basis data dari satu data ke data lainnya dan biasanya digunakan pada saat relasi (*relationship*) data

### 2.3.5 Normalisasi

Normalisasi adalah transformasi gambaran penggunaan yang kompleks dan data tersimpan ke sekumpulan bagian-bagian struktur data yang kecil dan stabil. Di samping menjadi lebih sederhana dan stabil, struktur data yang dinormalisasikan akan lebih mudah diatur dari pada struktur data lainnya. Normalisasi dilakukan guna memeriksa serta memperbaiki ranvangan logis yang telah dibuat sehinggah dapat memenuhi persyaratan yang dibutuhkan dan juga untuk menghindari adanya duplikasi data yang tidak penting. Selain itu normalisasi juga merupakan suatu teknik untuk memecahkan relasi-relasi yang memiliki anomali menjadi sutu relasi yang lebih kecil serta terstruktur dengan baik (*well-structured*) [5].

Berikut langkah dalam melakukan normalisasi data [5]:

#### 1. Bentuk Normalisasi Pertama (1NF)

Pada tahap pertama ini proses normalisasi yang dilakukan adalah menghilangkan semua kelompok-kelompok yang berulang dan mulai mengidentifikasi kunci utama. Pada proses ini hubungan perlu dipecahkan kedalam dua atau lebih hubungan. Ketika melakukan langkah pertama ini hubungan yang terbentuk mungkin sudah terbentuk normalisasi ketiga atau bahkan banyak tahap yang akan diperlukan untuk

mengubah hubungan tersebut kedalam bentuk normalisasi ketiga. Namun bentuk normalisasi ini hanya merupakan sifat dasar relasi dan belum tentu merupakan relasi yang terstruktur dengan baik. Pada saat melakukan bentuk normalisasi pertama ini harus dipastikan bahwa tidak ada atribut yang bernilai banyak (*multie value*).

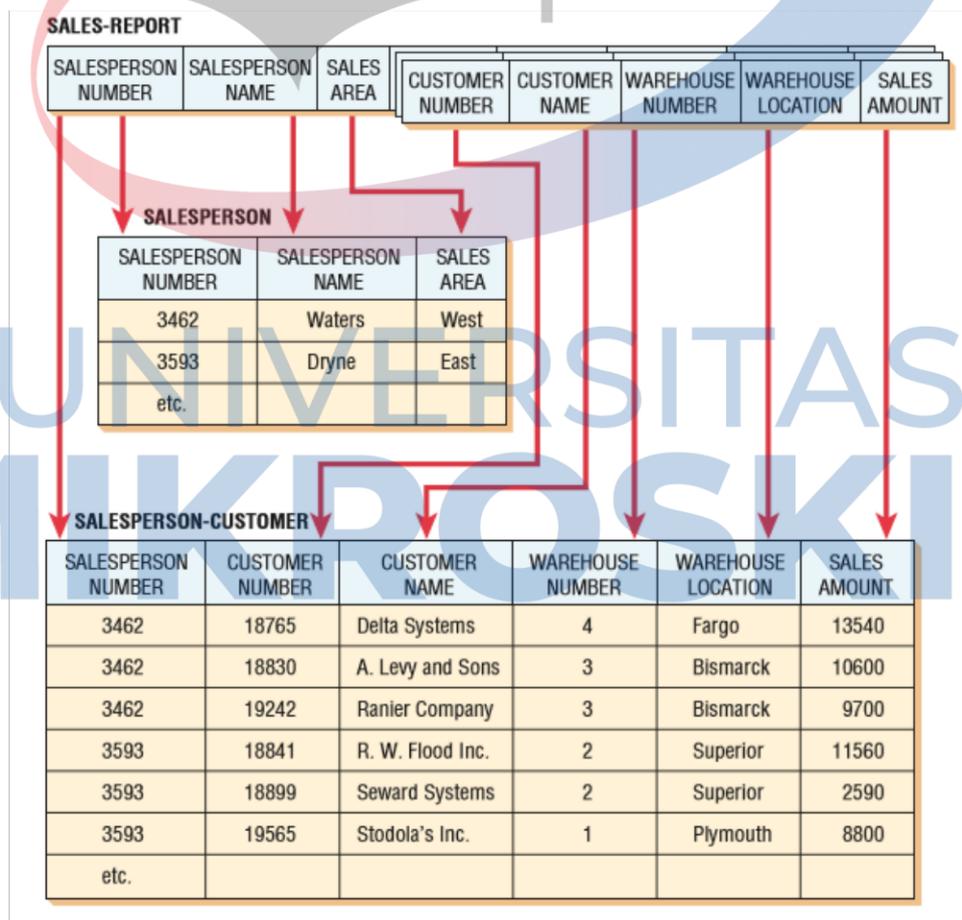
Relasi disebut juga sebagai 1NF jika memenuhi sebagai berikut

- Jika seluruh atribut dalam relasi bernilai atomic (*atomic value*).
- Jika seluruh atribut dalam relasi bernilai tunggal (*single value*).
- Jika relasi tidak memuat set atribut berulang.
- Jika semua *record* mempunyai sejumlah atribut yang sama.

Mengubah relasi UNF menjadi bentuk 1NF dapat dilakukan dengan cara sebagai berikut:

- Melengkapi nilai-nilai dalam atribut.
- Mengubah struktur relasi.

Berikut merupakan contoh bentuk Normalisasi Pertama (1NF):



**Gambar 2. 8** Contoh Penulisan Kamus Data

## 2. Bentuk Normalisasi Kedua (2NF)

Pada tahap kedua ini seluruh atribut-atribut secara fungsional akan bergantung pada kunci utama. Oleh karena itu proses selanjutnya adalah menghilangkan semua atribut yang bergantung secara parsial, kemudian menempatkannya ke hubungan yang lain.

Relasi disebut sebagai 2NF jika memenuhi kriteria sebagai berikut:

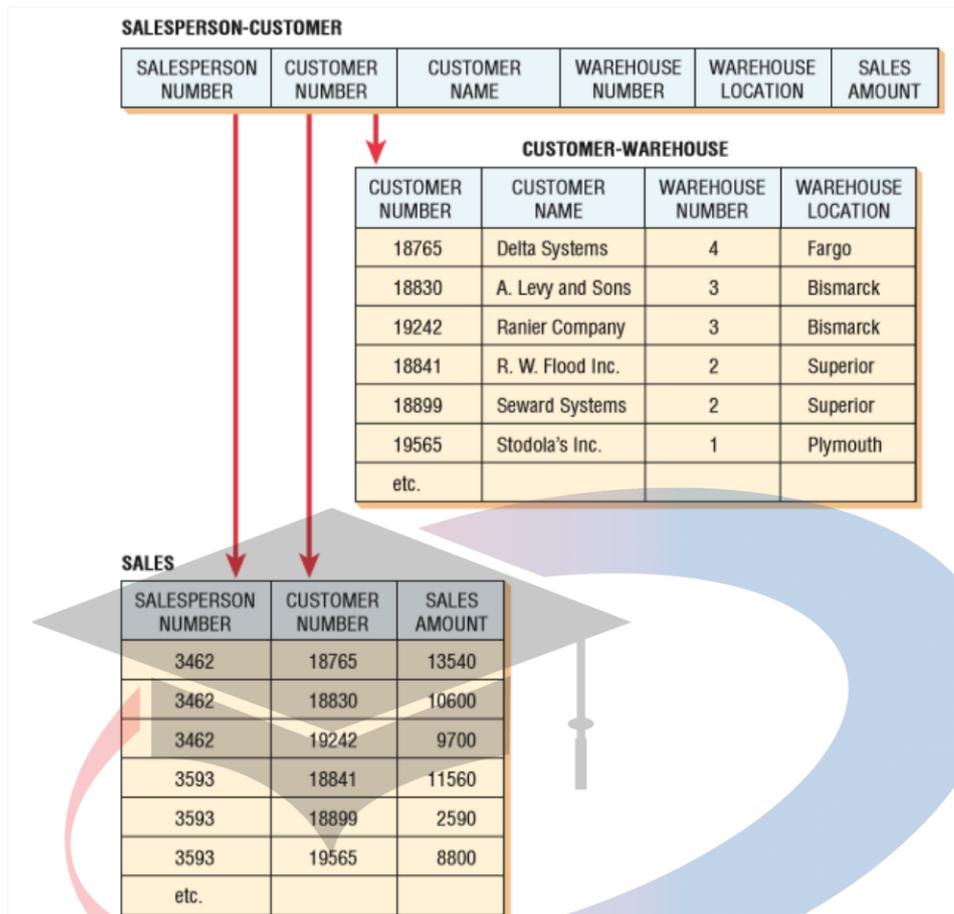
- a. Jika memenuhi kriteria 1NF
- b. Jika atribut nonkunci FD pada PK

Kriteria tersebut mengidentifikasi bahwa relasi 2NF menuntut telah didefinisikan atribut PK dalam relasi. Mengubah relasi 1NF menjadi 2NF dapat dilakukan dengan mengubah struktur relasi dengan cara:

- a. Identifikasi FD relasi 1NF (jika perlu gambarkan diagram ketergantungan datanya).
- b. Berdasarkan informasi tersebut, dekomposisi relasi 1NF menjadi relasi-relasi baru sesuai FD-nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak sebagai PK pada relasi baru.

Contoh bentuk Normalisasi Pertama (2NF):

UNIVERSITAS  
MIKROSKIL



**Gambar 2. 9** Contoh Normalisasi 2NF

### 3. Bentuk Normalisasi Ketiga (3NF)

Pada tahap ketiga ini hasil normalisasi suatu hubungan sudah terlihat dibentuk normal ketiga, jika seluruh atribut yang bukan merupakan kunci secara fungsional sepenuhnya bergantung pada kunci utama dan tidak ada lagi ketergantungan transitif. Dengan cara yang sama seperti pada tahap sebelumnya, ada kemungkinan untuk memecahkan kembali suatu hubungan menjadi dua hubungan.

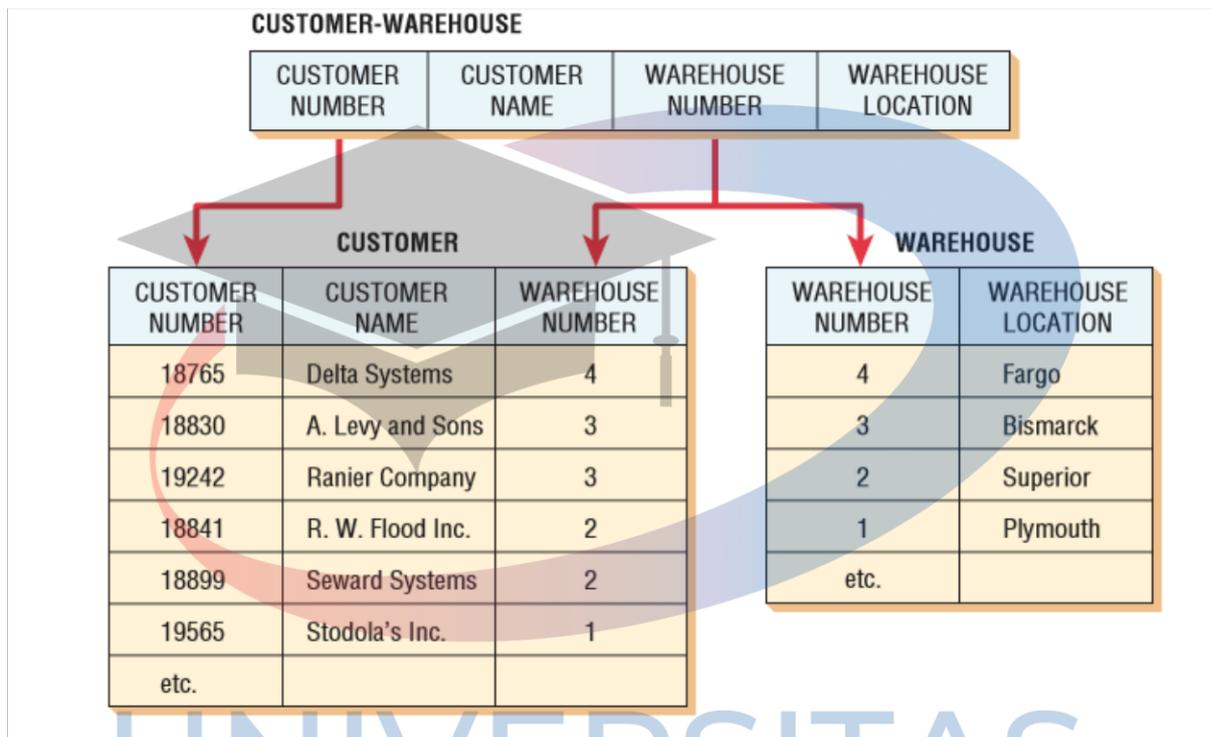
Suatu relasi disebut sebagai 3NF jika memenuhi kriteria berikut:

- a. Jika memenuhi kriteria 2NF.
- b. Jika setiap atribut nonkunci tidak TDF (*non transitive dependency*) terhadap PK.

Permasalahan dalam 3NF adalah keberadaan penentu yang tidak merupakan bagian dari PK menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai FK (duplikasi berbeda dengan karangkapan data). Mengubah relasi 2NF menjadi 3nf dapat dilakukan dengan mengubah struktur relasi dengan cara:

- Identifikasi TDF relasi 2NF (jika perlu gambarkan diagram ketergantungan datanya).
- Berdasarkan informasi tersebut, dekomposisi relasi 2NF menjadi relasi-relasi baru sesuai TDF nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak sebagai PK pada relasi baru.

Berikut contoh dari bentuk normalisasi ketiga (3NF).



**Gambar 2. 10** Contoh Normalisasi 3NF

### 2.3.6 PIECES

Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek, antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Analisis ini disebut dengan *Pieces Analysis (Performance, Information, Economy, Control, Efficiency, and Service)* [9].

*Framework Pieces* adalah kerangka yang dipakai untuk mengklasifikasikan suatu *problem, opportunities, dan directives* yang terdapat pada bagian *scope definition* perancangan sistem. Dengan kerangka ini, dapat dihasilkan hal-hal baru yang dapat menjadi pertimbangan dalam pengembangan sistem. Setiap huruf pada *Pieces* mempresentasikan sebuah kategori dalam dalam perumusan masalah yang ada, yaitu:

1. Analisis Kinerja (*Performance*)

Analisis kinerja adalah kemampuan menyelesaikan tugas pelayanan dengan cepat sehingga sasaran atau tujuan dapat tercapai. Kinerja diukur dengan jumlah produksi dan waktu tanggap (*Response Time*) dari suatu sistem.

## 2. Analisis Informasi (*Information*)

Analisis informasi adalah evaluasi kemampuan sistem informasi dalam menghasilkan nilai atau produk yang bermanfaat untuk menyikapi peluang dalam menangani masalah yang muncul. Situasi dalam analisis informasi ini meliputi:

- a. Akurasi, informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan.
- b. Relevan, informasi tersebut memiliki manfaat bagi pihak pemakai maupun pihak pengelola. Relevansi setiap orang berbeda satu dengan yang lainnya.

## 3. Analisis Ekonomi (*Economy*)

Analisis ekonomi adalah penilaian sistem atas biaya dan keuntungan yang akan didapatkan dari sistem yang diterapkan. Sistem ini memberikan penghematan operasional dan keuntungan bagi instansi atau perusahaan. Analisis ini meliputi biaya dan keuntungan.

## 4. Analisis Keamanan (*Controlling*)

Analisis Keamanan adalah penilaian terhadap sistem keamanan keamanan yang digunakan harus dapat mengamankan data dari kerusakan, misalnya dengan melakukan *back-up* data. Selain itu sistem keamanan juga harus dapat mengamankan data dari akses yang tidak diizinkan. Analisis ini meliputi pengawasan dan pengendalian.

## 5. Analisis Efisiensi (*Efficiency*)

Analisis efisiensi adalah sumber daya yang berguna meminimalkan pemborosan.

Efisiensi dari sistem yang dikembangkan adalah pemakaian secara maksimal terhadap sumber daya infrastruktur, dan sumber daya manusia serta menganalisis keterlambatan pengolahan data yang terjadi.

## 6. Analisis Layanan (*Services*)

Analisis layanan adalah mengkoordinasikan aktifitas dalam pelayanan yang ingin dicapai sehingga tujuan dan sasaran pelayanan dapat dicapai.