BAB II

LANDASAN TEORI

2.1. Pengertian Sistem

Pengertian Sistem dalam pengertian yang paling umum adalah *sekumpulan benda yang memiliki hubungan di antara mereka*. Kata sistem sendiri berasal dari bahasa Latin (systēma) dan bahasa Yunani (sustēma) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

Sedangkan menurut para ahli, Sistem diartikan sebagai berikut :

1. **LUDWIG VON BARTALANFY**

   Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.

2. **ANATOL RAPOROT**

   Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

3. **L. ACKOF**

   Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Mengacu pada beberapa definisi sistem di atas, dapat juga diartikan, sistem adalah sekumpulan unsur / elemen yang saling berkaitan dan saling

mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Sebagai contoh, dalam sistem komputer terdapat software (perangkat lunak), hardware (perangkat keras), dan brainware (sumber daya manusia).

2.1.1. Elemen Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat. Sistem merupakan kumpulan bagian-bagian atau sub-sub sistem yang disatukan dan dirancang untuk mencapai suatu tujuan. Tujuan dari sistem tersebut adalah untuk mengorganisasikan sistem informasi yang baru agar dapat mengatasi berbagai masalah yang terjadi pada suatu organisasi, serta memberikan pengertian mengenai suatu bentuk sistem yang ada pada suatu organisasi serta trik-trik manageman yang berkaitan dengan sistem informasi manageman (SIM) berbasis komputer.

2.1.2. Karakterisitik Sistem

Sistem mempunyai beberapa karakteristik atau sifat tertentu, antara lain :

- Komponen sistem (Component) : Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang saling bekerja sama membentuk suatu komponen sistem
- Batasan sistem (Boundary) : Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

- Sub sistem (Sub system) : Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasaran masing-masing.
- Lingkungan luar sistem (Encironment) : Suatu sistem yang ada di luar dari batas sistem yang dipengaruhi oleh operasi sistem.
- Penghubung sistem (Interface) : Media penghubung antara suatu sub sistem dengan sub sistem yang lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu sub sistem ke sub sistem lainnya
- Masukan sistem (Input) : Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukan supaya sistem tersebut dapat berinteraksi.
- Keluaran sistem (Output) : Hasil energi yang diolah dan diklasifikasi menjadi keluaran yang berguna dan sisa pembuangan.
- Pengolahan sistem (Proses : Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan mengubah masukan menjadi keluaran.
- Sasaran sistem (Object) : Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.

2.1.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya :

1. Sistem diklasifikasikan sebagai sistem abstrak (abstract system) dan sistem phisik (physical system).

- Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara phisik. Misalnya sistem teologia , yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan tuhan.
- Sistem phisik merupakan sistem yang ada secara phisik misalnya sistem komputer , sistem akuntasi, sistem pruduksi, dan lain sebagainnya.

2. Sistem diklasifikasikan sebagai sistem alamiah (natural system) dan sistem buatan manusia (human made system).
   - Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi ,
   - Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan human-machine system atau ada yang menyebut dengan man-machine system, karena menyangkut penggunaan komputer yang ber-interaksi dengan manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (deterministic system) dan sistem tak tentu (probabilistic system).
   - Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti. Sehingga interaksi antara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat

diramalkan. Sistem komputer adalah contoh dari sistem tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan.

- Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (closed system) dan sistem terbuka (open system).
   - Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada system yang benar-benar tertutup. Yang ada hanyalah relatively closed system (secara relatip tertutup tidak benar-benar terbuka).
   - Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar atau subsistem yang lainnya. Karena sistem sifat terbuka dan terpengaruh oleh lingkungan luarnya. Maka suatu sistem harus mempunyai suatu sistem pengendali yang baik.

2.2. Pengertian Informasi

Informasi merupakan hasil proses data-data yang beragam yang telah dibentuk sedemikan rupa sehingga sesuai dengan permintaan pengguna, tentunya telah mengalami proses yang telah tersusun dengan baik dan benar dengan kriteria yang diinginkan. Informasi merupakan hal yang sangat penting bagi manajemen

didalam pengambilan keputusan. Informasi adalah data yang telah diolah dan bermanfaat bagi yang membutuhkannya. Sumber dari informasi adalah data. Data adalah kumpulan angka maupun karakter yang mempunyai arti tertentu dan diambil dari realita atau kenyataan yang terjadi pada suatu lokasi

Informasi mempunyai ciri-ciri sebagai berikut:

1. Data yang telah diolah.
2. Menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima.
3. Menggambarkan suatu kejadian dan kesatuan nyata.
4. Digunakan untuk mengambil keputusan

Penghubung sistem merupakan penghubung antara satu subsistem dengan subsistem yang lainnya yang memungkinkan adanya sumber-sumber daya mengalir dari satu sub sistem dengan sub sistem yang lainnya.

1. Masukkan Sistem (Input)

   Masukkan sistem adalah energi yang dimasukkan ke dalam sistem yang dapat berupa masukkan perawatan (Maintenance Input) dan masukkan sinyal (Signal Input).

2. Keluaran Sistem (Output)

   Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat berupa masukan dari subsistem yang lain atau kepada supra sistem

3. Pengolah Sistem (Process)

   Pengolah sistem akan mengubah masukan menjadi keluaran.

4. Sasaran Sistem (Objectives)

   Sasaran sistem sangat menentukan sekali masukkan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuan.

2.3. Pengertian Sistem Informasi

Sistem Informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, mengolah, dan menyimpan data dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

2.4. Metode Pendekatan dan Pengembangan Sistem

2.4.1. Metode Pendekatan Sistem

2.4.1.1. Pemecahan Masalah

Masalah merupakan suatu kondisi yang memiliki potensi untuk menimbulkan kerugian luar biasa atau menghasilkan keuntungan luar biasa. Jadi pemecahan masalah berarti tindakan memberikan respon terhadap masalah untuk menekan akibat buruknya atau memanfaatkan peluang keuntungannya. Oleh karena itu masalah penting untuk dipecahkan. Keputusan adalah pemilihan suatu strategi atau tindakan. Pengambilan keputusan adalah tindakan memilih strategi atau aksi yang manajer yakini akan memberikan solusi terbaik atas masalah tersebut.

Elemen-elemen proses pemecahan masalah :

- Elemen-elemen Standar : menggambarkan keadaan yang diharapkan apa yang harus dicapai oleh sistem.
- Informasi : menggambarkan keadaan saat ini – apa yang sedang dicapai oleh sistem.

Perbedaan antara masalah dan gejala. Gejala adalah kondisi yang dihasilkan oleh masalah. Untuk memberikan ilustrasi ini, kita ambil contoh, seorang manajer dihadapkan pada suatu gejala seperti laba yang rendah. Dalam hal ini ada masalah penyebab laba rendah. Jadi dalam kaitan ini, masalah adalah penyebab dari suatu persoalan, atau penyebab dari suatu peluang.

Jenis-jenis masalah :

- Masalah terstruktur; apabila terdiri dari elemen dan hubunganhubungan antar elemen yang semuanya dipahami oleh pemecah masalah.
- Masalah tak terstruktur; berisi elemen-elemen atau hubungan antar elemen yang tidak dipahami oleh pemecah masalah.
- Masalah semi-terstruktur, masalah yang berisi sebagian elemenelemen atau hubungannya yang dimengerti oleh pemecah masalah.

2.4.1.2. Pendekatan Sistem

Proses pemecahan masalah secara sistematis bermula dari John Dewey, seorang professor filosofi di Columbia University pada awal abad

ini. Dalam bukunya 1910 diidentifikasi ada 3 seri penilaian dalam memecahkan suatu kontroversi yang memadai :

1) Mengenali kontroversi.
2) Menimbang klaim alternatif.
3) Membentuk penilaian.

Tahap-tahap dan langkah-langkah pendekatan sistem :

Tahap I : Usaha Persiapan Langkah

a) Memandang perusahaan sebagai suatu sistem.
b) Mengenali sistem lingkungan.
c) Mengidentifikasi subsistem perusahaan.

Tahap II : Usaha Definisi Langkah

a) Bergerak dari tingkat sistem ke subsistem.
b) Menganalisa bagian sistem dalam urutan tertentu.

Tahap III : Usaha Solusi Langkah

a) Mengidentifikasi solusi alternatif.
b) Mengevaluasi solusi alternatif.
c) Memilih solusi terbaik.
d) Menerapkan solusi terbaik.
e) Membuat tindak lanjut bahwa solusi itu efektif.

Tiap tingkatan manajemen adalah suatu sub-sistem.

- Arus informasi
- Arus keputusan

Faktor-faktor Pribadi yang Mempengaruhi

Pemecahan Masalah :

Tiap manajer memiliki gaya pemecahan masalah yang unik. Gaya ini mempengaruhi keterlibatannya dalam merasakan masalah, mengumpulkan informasi, dan menggunakan informasi.

Merasakan masalah :

Ada tiga kategori dasar dalam gaya merasakan masalah *(problemsensing style)*, yaitu :

1) Menghindar masalah *(problem avoider)*, mengambil sikap positif dan menganggap bahwa semua baik-baik saja.
2) Pemecah masalah *(problem solver)*, manajer ini tidak mencari dan tidak menghalangi masalah.
3) Pencari masalah *(problem seeker)*, manajer ini menikmati pemecahan masalah dan mencarinya.

Mengumpulkan informasi :

Gaya mengumpulkan informasi ada dua :

1) Gaya teratur *(preceptive styles)*, manajer jenis ini mengikuti *management by exception* dan menyaring segala sesuatu yang tidak berhubungan dengan area minatnya.
2) Gaya menerima *(receptive style)*, manajer jenis ini ingin melihat semuanya, kemudian menentukan apakah informasi tersebut bernilai baginya atau orang lain dalam organisasi.

Menggunakan informasi :

Manajer juga cenderung menggunakan salah satu dari dua gaya menggunakan informasi, yaitu :

1) Gaya sistematik *(systematic style)*. Manajer memberi perhatian khusus untuk mengikuti suatu metode yang telah ditetapkan, misalnya pendekatan sistem.
2) Gaya intuitif *(intuitive style)*.Manajer tidak lebih menyukai suatu metode tertentu tetapi menyesuaikan pendekatan dengan situasi. Di dalam pendekatan sistem, tiga komentar berikut perlu diperhatikan yaitu :
   a) Pendekatan sistem sebena rnya hanyalah akal sehat *(common sense)*.
   b) Pendekatan sistem hanyalah satu cara memecahkan masalah.
   c) Pendekatan sistem adalah metodologi sistem dasar. Suatu metodologi adalah suatu cara yang telah ditetapkan untuk melaksanakan sesuatu.

2.4.2. Metode Pengembangan Sistem

2.4.2.1 Waterfall Model

Metode ini merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan

pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisa, Design, Code dan Testing, Penerapan dan Pemeliharaan.

1. **Analisa**

   Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa malakukan sebuah penelitian, wawancara atau study literatur. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirment* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menterjemahkan ke dalam bahasa pemprogram.

2. **Design**

   Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software*

requirment. Dokumen inilah yang akan digunakan *proggrammer* untuk melakukan aktivitas pembuatan sistemnya.

3. Coding & Testing

Coding merupan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan meterjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

1) Penerapan

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

2) Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

2.4.2.2 Incremental Model

Model incremental (Incremental waterfall model) merupakan perbaikan dari model waterfall dan sebagai standar pendekatan top-down. Ide dasar dari model ini adalah membangun software secara meningkat (increment) berdasarkan kemampuan fungsional. Model incremental ini diaplikasikan pada sistem pakar dengan penambahan rules yang mengakibatkan bertambahnya kemampuan fungsional sistem. Keuntungan dari model ini adalah bahwa penambahan kemampuan fungsional akan lebih mudah diuji, diverifikasi, dan divalidasi dan dapat menurunkan biaya yang dikeluarkan untuk memperbaiki sistem. Model incremental merupakan model continous rapid prototype dengan durasi yang diperpanjang hingga akhir proses pengembangan. Pada model prototipe biasa, prototipe hanya dibuat pada tahap awal untuk mendapatkan kebutuhan user.

2.4.2.3 Code-and-Fix Model

Model ini mengembangkan software dengan cara membuat program dan kemudian diperbaiki jika terdapat kesalahan. Model ini merupakan model awal yang digunakan untuk mengembangkan software. Namun sejak tahun 1970-an, model ini mulai ditinggalkan dan dikembangkan model waterfall yang memberikan metodologi lebih sistematik dan sangat membantu terutama pada proyek-proyek yang besar. Namun kesulitan pada model waterfall adalah perlu adanya informasi yang lengkap pada setiap tahapnya, dan bukan sesuatu hal yang

mudah untuk mendapatkan informasi tersebut. Pada prakteknya, sering tidak mungkin untuk menulis dokumentasi kebutuhan yang lengkap sebelum dibangun prototipe. Sehingga yang terjadi adalah "kerja dua kali", membuat prototipe, kemudian dari prototipe diperoleh informasi kebutuhan dan barulah dibangun sistem final.

2.4.2.4 Prototyping Model

Metode ini sering digunakan pada dunia *riil*. Karena metode ini secara keseluruhan akan mengacu kepada kepuasan *user*. Bisa dikatakan bahwa metode ini merupakan metode *waterfall* yang dilakukan secara berulang-ulang.

1. **Tahapan metode prototyping :**

   - **Pemilihan Fungsi.**

     Mengacu pada pemilahan fungsi yang harus ditampilkan oleh prototyping. Pemilahan harus selalu dilakukan berdasarkan pada tugas-tugas yang relevan yang sesuai dengan contoh kasus yang akan diperagakan.

   - **Penyusunan Sistem Informasi**

     Bertujuan memenuhi permintaan kebutuhan akan tersedianya prototype.

   - **Evaluasi.**

   - **Penggunaan selanjutnya.**

2. Jenis Jenis Prototyping

- **Feasibility prototyping**

  Digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk system informasi yang akan disusun.

- **Requirement prototyping**

  Digunakan untuk mengetahui kebutuhan aktivitas bisnis user. Misalnya dalam sebuah perusahaan terdapat *user* direktur, manajer, dan karyawan. Maka penggunaan sistem dapat dibedakan berdasarkan *user* tersebut sesuai dengan kebutuhannya.

- **Desain Prototyping**

  Digunakan untuk mendorong perancangan system informasi yang akan digunakan.

- **Implementation prototyping**

  Merupakan lanjutan dari rancangan protipe, prototype ini langsung disusun sebagai suatu system informasi yang akan digunakan.

2.4.2.5 Agile Model

Pada dekade 90-an diperkenalkan metodologi baru yang dikenal dengan nama *agile methods*. Metodologi ini sangat revolusioner perubahannya jika dibandingkan dengan metode sebelumnya. *Agile Methods* dikembangkan karena pada metodologi tradisional terdapat banyak hal yang membuat proses pengembangan tidak dapat berhasil dengan baik sesuai tuntutan *user*. Saat ini metodologi ini sudah cukup banyak berkembang, diantaranya adalah:

1) eXtreme Programming (XP)

2) Scrum Methodology

3) Crystal Family

4) Dynamic Systems Development Method (DSDM)

5) Adaptive Software Development (ASD)

6) Feature Driven Development (FDD)

Jika kita lihat, *agile* bisa berarti tangkas, cepat, atau ringan. *Agility* merupakan metode yang ringan dan cepat dalam pengembangan perangkat lunak. *Agile Alliance* mendefinisikan 12 prinsip untuk mencapai proses yang termasuk dalam *agility*:

- Prioritas tertinggi adalah memuaskan pelanggan melalui penyerahan awal dan perangat lunak yang bernilai.
- Menerima perubahan *requirements* meskipun perubahan tersebut diminta pada akhir pengembangan.
- Memberikan perankat lunak yang sedangdikerjakan dengan sering,beberapa min ggu atau bulan, dengan pilihan waktu yang paling singkat.
- Pihak bisnis dan penggembangan harus berkerja sama setiap hari selam penggembangan berjalan.

- Bangun proyek dengan individu-individu yang bermotivasi tinggi dengan memberikan lingkungan dan dukungan yang diperlukan, dan mempercyai mereka sepenuhnya untuk menyelesaikan pekerjaannya.
- Metode yang paling efektif dan efisien dala m menyampaikan informasi kepada tim pengembangan adalah dengan komunikasi langsung *face to face*.
- Perangkat lunak yang dikerjakan merupakan pengukur utama kemajuan.
- Proses agile memberikan proses pengembangan yang bias ditopang.Sponsor,pengembangan, dan *user* garus bias menjaga ke-konstanan langkah yang tidak pasti.
- Perhatian yang harus terus mennerus terhadap rangcangan dan teknik yang baik meningkatkan *agility*.
- Keserdahanaan seni untuk meminimalkan jumlah pekerjaan adalah penting.
- Arsitektur,*requirements*, dan rancangan terbaik muncul dari tim yang mengatur sendiri.
- Pada interval reguler tertentu,tim merefleksikan bagaimana menjadi lebih efektif,kemudian menyesuaikannya.

Kelebihan Metode Agile

- Meningakatkan rasio kepuasan pelanggan.
- Bisa melakukan reviw pelanggan mengenai software yang dibuat lebih awal.

- Mengurangi resiko kegagalan implementasi software dari non-teknis.
- Besar kerugian baik secara material atau imaterial tidak terlalu besar jika terjadi kegagalan.

2.4.3. Alat Bantu Analisis

2.4.3.1 Flow Map

Flowmap adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program.

Cara - cara membuat flowmap

Seorang analis dan programmer akan membuat flowmap ada beberapa petunjuk yang harus diperhatikan, seperti :

1. Flowmap digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan MENGHITUNG PAJAK PENJUALAN.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada

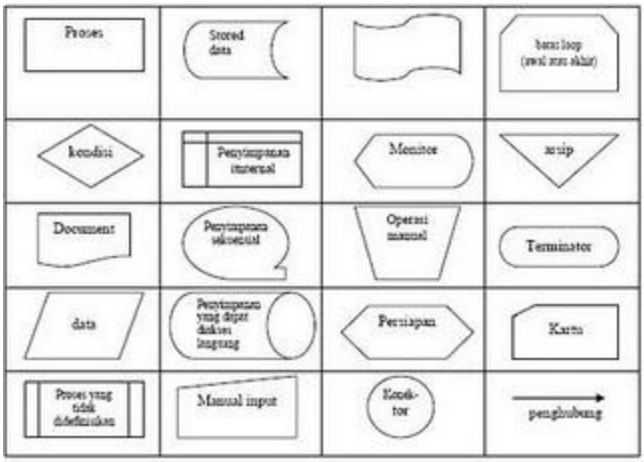
flowmap yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

7. Gunakan simbol-simbol flowmap yang standar.

Jenis - jenis Flowmap

1. Flowmap Sistem
2. Flowmap Paperwork atau Flowmap Dokumen
3. Flowmap Skematik
4. Flowmap Program
5. Flowmap Proses

Symbol Flowmap



Symbol 2.1 Flow Map

2.4.3.2 Diagram Kontek

Model berikutnya menjawab sejumlah pertanyaan yang muncul dalam pembuatan statement of purpose. Context Diagram merupakan kejadian tersendiri dari suatu diagram alir data. Dimana satu lingkaran

merepresentasikan seluruh sistem. Context Diagram ini harus berupa suatu pandangan, yang mencakup masukan-masukan dasar, sistem-sistem dan keluaran. Context Diagram merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran daa menuju dan dari sistem diketahui penganalisis dari wawancara dengan user dan sebagai hasil analisis dokumen.

Context diagram menggarisbawahi sejumlah karakteristik penting dari suatu sistem:

- Kelompok pemakai, organisasi, atau sistem lain dimana sistem kita melakukan komunikasi yang disebut juga sebagai terminator.
- Data dimana sistem kita menerima dari lingkungan dan harus diproses dengan cara tertentu.
- Data yang dihasilkan sistem kita dan diberikan ke dunia luar.
- Penyimpanan data yang digunakan secara bersama antara sistem kita dengan terminator. Data ini dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya,, dibuat oleh lingkungan dan digunakan oleh sistem kita.
- Batasan antara sistem kita dan lingkungan.

Context Diagram dimulai dengan penggambaran terminator, aliran data, aliran kontrol penyimpanan, dasn proses tunggal yang menunjukkan keseluruhan sistem. Bagian termudah adalah menetapkan proses (yang hanya terdiri dari satu lingkaran) dan diberi nama yang mewakili sistem. Nama dalam hal ini dapat menjelaskan proses atau pekerjaan atau dalam kasus ekstrim berupa nama perusahaan yang dalam hal ini mewakili proses yang dilakukan keseluruhan organisasi.

Terminator ditunjukkan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau penyimpanan eksternal Antar terminator tidak diperbolehkan komunikasi langsung. Pada kenyataannya hubungan antar terminator dilakukan, tetapi secara definitif karena terminator adalah bagian dari lingkungan, maka tidak relevan jika dibahas dalam context diagram.

Context Diagram memiliki aturan sebagai berikut:

a. Jika terdapat banyak terminator yang mempunyai banyak masukan dan keluaran diperbolehkan untuk digambarkan lebih dari satu kali sehingga mencegah penggambaran yang terlalu rumit, dengan ditandai secara khusus untuk menjelaskan bahwa terminator yang dimaksud adalah identik. Tanda tersebut dapat berupa asterik (*) atau pagar (#).
b. Jika terminator mewakili individu sebaiknya diwakili oleh peran yang dimainkan personil tersebut. Alasan pertama adalah personil yang berfungsi untuk melakukan itu dapat berganti sedang Context

Diagram harus tetap akurat walaupun personil berganti. Alasan kedua adalah seorang personil dapat memainkan lebih dari satu peran.

Karena fokus uitama adalah mengembangkan model, maka penting untuk membedakan sumber (*resource*) dan pelaku (*handler*)., pelaku adalah mekanisme, perangkat atau media fisik yang mentransportasikan data ke/dari sistem, karena pelaku seringkali familier dengan pemakai dalam implementasi sistem berjalan, maka sering menonjol sebagai sesuatu yang harus digambarkan lebih dari sumber data itu sendiri. Sedangkan sistem baru dengan konsep pengembangan teknologinya membuat pelaku menjadi sesuatu yang tidak perlu digambarkan.

Aliran dalam context diagram memodelkan masukan ke sistem dan keluaran dari sistem seperti halnya sinyal kontrol yang diterima atau dibuat sistem. Aliran data hanya digambarkan jika diperlukan untuk mendeteksi kejadian dalam lingkungan dimana sistem harus memberikan respon atau membutuhkan data untuk menghasilkan respon. Selain itu, aliran data dibutuhkan untuk menggambarkan transportasi antara sistem dan terminator. Dengan kata lain aliran data digambarkan jika data tersebut diperlukan untuk menghasilkan respon pada kejadian tertentu. Dalam hal ini kita seharusnya menggambar context diagram dengan asumsi bahwa masukan disebabkan dan diawali oleh terminator, sedangkan keluaran disebabkan dan diawali oleh sistem.

Dengan mencegah interaksi yang tidak perlu (*extraneous prompts*) yang berorientasi pada implementasi masukan-keluaran dan mengkonsentrasikan pemodelan pada jaringan aliran data. Kadang-kadang diperlukan dialog karena terminator tidak tahu sistem memerlukan masukan atau sistem tidak memberikan keluaran karena tidak tahu terminator membutuhkannya. Dalam hal ini interaksi menjadi diperlukan dan diasumsikan menjadi bagian esensi sistem

2.4.3.1 Data Flow Diagram

Empat simbol dasar yang digunakan untuk memetakan gerakan diagram aliran data adalah:

a. External Entity (Entitas)/terminator

Suatu entitas dapat disimbolkan dengan suatu notasi kotak sebagai berikut:

Kotak ini digunakan untuk menggambarkan suatu entitas eksternal (bagian lain, sebuah perusahaan, seseorang atau sebuah mesin) yang dapat mengirim data atau menerima data dari sistem. Entitas ini disebut juga

Sumber atau tujuan data, dan dianggap eksternal terhadap sistem yang sedang digambarkan. Setiap entitas diberi label dengan sebuah nama yang sesuai. Meskipun berinteraksi dengan sistem, namun dianggap di luar batas-batas sistem. Entitasentitas tersebut harus diberi nama dengan suatu kata benda entitas yang sama bisa digunakan lebih dari sekali atas suatu diagram aliran data tertentu untuk menghindari persilangan antara jalur-

jalur aliran data Bentuk dari eksternal entity diantaranya adalah sebagai berikut:

- Suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang sedang dikembangkan.
- Orang/sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan
- Suatu organisasi atau orang yang berada di luar organisasi seperti misalnya langganan, pemasok, dll.
- Sistem informasi yang lain di luar sistem yang sedang dikembangkan
- Sumber asli dari suatu transaksi Penerima akhir dari suatu laporan yang dihasilkan oleh sistem

b. Data Flow/Arus data

Suatu arus data dapat disimbolkan dengan menggunakan suatu notasi tanda panah berikut .

Symbol 2.2 Arus Data

Tanda panah ini menunjukkan perpindahan data dari satu titik ke titik yang lain, dengan kepala tanda panah mengarah ke tujuan data. Karena sebuah tanda panah menunjukkan seseorang, tempat atau sesuatu, maa harus digambarkan dalam kata benda

Bentuk dari arus data diantaranya adalah sebagai berikut:

- Formulir atau dokumen yagn digunakan di perusahaan
- Laporan tercetak yang dihasilkan oleh sistem

- Tampilan atau output di layar komputer yang dihasilkan oleh sistem
- Masukan untuk komputer
- Komunikasi ucapan
- Surat-surat atau memo
- Data yang dibaca atau direkamkan ke suatu file
- Suatu isian yang dicatat pada buku agenda
- Transmisi data dari suatu komputer ke komputer yang lain

c. Process/Proses

Bujur sangkar dengan sudut membulat/lingkaran digunakan untuk menunjukkan adanya proses transformasi. Proses-proses tersebut selalu menunjukkan suatu perubahan dalam di dalam atau perubahan data; jadi, aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk. Proses-proses yang menunjukkan hal itu di dalam sistem dan harus diberi nama menggunakan salah satu format berikut ini. Sebuah nama yang jelas memudahkan untuk memahami proses apa yang sedasng dilkukan. 1. Menetapkan nama sistem secara keseluruhan saat menamai proses pada level yang lebih tinggi. Contoh: *sistem kontrol inventaris*.

Menamai suatu subsistem utama, menggunakna nama-nama seperti : *Sistem pelaporan inventaris* atau *Sistem pelayanan konsumen internet*.

Menggunakan format kata kerja – kata sifat – kata benda untuk proses-proses yang mendetail. Kata kerja yang menggambarkan jenis

kegiatn yang seperti ini, misalnya *menghitung*, *memverifikasi*, *menyiapkan*, *mencetak* atau *menambahkan*. Contoh-contoh nama proses yang lengkap adalah: *menghitung pajak penjualan*, *memverifikasi status rekening konsumen*, *menyiapkan invoice pengapalan*, *mencetak laporan yang diurutkan ke belakang*, *mengirim konfirmasi email ke konsumen*, *memverifikasi neraca kartu kredit* dan *menambah record inventaris*.

d. Data Store (Penyimpanan Data)

Data store dilambangkan dengan bujur sangkar dengan ujung terbuka. yang menunjukkan penyimpanan data. Penyimpanan data menandakan penyimpanan manual, seperti lemari file/sebuah file/basisdata terkomputerisasi. Karena penyimpanan data mewakili seseorang, tempat atau sesuatu, maka diberi nama dengan sebuah kata benda. Penyimpanan data sementara seperti kertas catatan/sebuah file komputer sementara tidak dimasukkan ke dalam diagram aliran data.

Bentuk dari penyimpanan data diantaranya adalah sebagai berikut:

- Suatu file atau database di sistem komputer
- Suatu arsip atau catatan manual
- Suatu kotak tempat data di meja seseorang
- Suatu tabel acuan manual
- Suatu agenda atau buku

Dalam penggambaran penyimpanan data perlu diperhatikan beberapa hal, antara lain:

a. Hanya proses saja yang berhubungan dengan simpanan data, karena yang mengggunakan/ merubah data di simpanan data adalah suatu proses.
b. Arus data yang menuju ke simpanan data dari suatu proses menunjukkan proses update terhadap data yang tersimpan di simpanan data. Update dapat berupa proses:
    - Menambah/menyimpankan record baru atau dokumen baru ke dalam simpanan data.
    - Menghapus record atau mengambil dokumen dari simpanan data.
    - Merubah nilai data di suatu record atau di suatu dokumen yang ada di simpanan data
c. Arus data yang berasal dari simpanan data ke suatu proses menunjukkan bahwa proses tersebut menggunakan data yang ada di simpanan data.
d. Untuk suatu proses yang melakukan kedua-duanya yaitu menggunakan dan update simpanan data dapat dipilih salah satu penggambaran.

Untuk menghindari garis arus data yang saling berpotongan sehingga membuat gambar di DFD menjadi ruwet, maka simpanan data/kesatuan luar dapat digambar lebih dari sebuah.

2.4.3.3 Kamus Data

Kamus data atau disebut juga dengan istilah systems data dictionary adalah katalog fakta tentang data dan kebutuhan kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, kamus data dapat digunakan sebagai alat komunikasi antara analis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporau-laporan dan database. Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja.

Keterangan lebih lanjut tentang struktur dari suatu arus data di DFD secara lebih terinci dapat dilihat di kamus data.

Isi Kamus Data

Apa yang perlu dicatat di kamus data? Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka kamus data harus memuat hal-hal berikut ini.

1. Nama arus data.

Karena kamus data dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di kamus data, sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DAD dapat langsung mencarinya dengan mudah di kamus data

2. Alias

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya. Misalnya bagian pembuat faktur dan langganan menyebut bukti penjualan.sebagai faktur, sedang bagian gudang menyebutnya sebagai tembusan permintaan persediaan. Balk faktur dan tembusan permintaan persediaan ini mempunyai struktur data yang sama, tetapi mempunyai struktur yang berbeda.

3. Bentuk data.

Telah diketahui bahwa arus data dapat mengalir :

- dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir;
- hasil dari suatu proses ke kesatuan luar, data yang mengalir ini biasanya terdapat di media laporan atau query tampilan layar atau dokumen hasil cetakan komputer;

- hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya dalam bentuk variabel atau parameter yang dibutuhkan bleb proses penerimanya;
- hasil suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variabel;
- dari simpanan data dibaca oleh suatu proses, data yang mengalir ini biasanya berupa suatu field (item data).

Dengan demikian bentuk dari data yang mengalir dapat berupa:

- dokumen dasar atau formulir;
- dokumen hasil cetakan computer
- laporan terecetak;
- tampilan di layar monitor;
- variabel;
- parameter;
- field.

Bentuk dari data ini perlu dicatat di kamus data, karena dapat digunakan untuk mengelompokkan kamus data ke dalam kegunaannya sewaktu perancangan sistem. Kamus data yang mencatat data yang mengalir dalam bentuk dokumen dasar atau formulir akan digunakan untuk merancang bentuk input sistem. Kamus data yang mencatat data yang mengalir dalam bentuk laporan tercetak dan dokumen hasil cetakan komputer akan digunakan untuk merancang output yang akan dihasilkan oleh sistem. Kamus data yang mencatat data yang mengalir dalam bentuk

tampilan di layar monitor akan digunakan juga untuk merancang tampilan layar yang akan dihasilkan oleh sistem. Kamus data yang mencatat data yang mengalir dalam bentuk parameter dan variabel akan digunakan untuk merancang proses dari program. Kamus data yang mencatat data yang mengalir dalam bentuk dokumen, formulir, laporan, dokumen cetakan komputer, tampilan di layar monitor, variabel dan field akan digunakan untuk merancang database.

4. Arus data.

Arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju. Keterangan arus data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini di DFD.

5. Penjelasan.

Untuk lebih memperjelas lagi tentang Anakna dari arus data yang dicatat di kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan ten-tang arus data tersebut. Sebagai misalnya nama dari arus data adalah TEMBUSAN PERMINTAAN PERSEDIAAN, maka dapat lebih dijelaskan sebagai tembusan dari faktur penjualan untuk meminta barang dari gudang.

6. Periode.

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di Kamus data karena dapat digunakan untuk mengidentifikasikan kapan input data harus dimasukkan ke sistem, kapan

proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

7. Volume

Volume yang perlu dicatat di kamus data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak. Volume ini digunakan untuk mengidentifikasikan besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah dari alat input, alat pemroses dan alat output.

8. Struktur data.

Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

2.4.3.4 Perancangan Basis Data

a. Normalisasi

Suatu teknik untuk mengorganisasi data ke tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi.

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan.

- Suatu teknik untuk mengorganisasi data ke tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi.
- Proses pengelompokan elemen data menjadi tabel yang menunjukkan entity sekaligus relasinya.

Menurut pendapat lain

- Istilah Normalisasi berasal dari E. F. Codd, salah seorang perintis teknologi basis data. selain dipakai sebagai metodologi tersendiri untuk menciptakan struktur tabel (relasi) dalam basis data (dengan tujuan utnuk mengurangi kemubaziran data) , normalisasi terkadang hanya diipakai sebagai perangkat verifikasi terhadap tabel-tabel yang dihasilkan oleh metodologi lain (misalnya E-R). Normalisasi memberikan panduan yang sangat membantu bagi pengembang untuk mencegah penciptaan struktur tabel yang kurang fleksibel atau mengurangi keflekxibelan.
- Kroenke mendefinisikan normalisasi sebagai proses untuk mengubah suatu relasi yang memiliki masalah tertentu ke dalam dua buah relasi atau lebih yang tidak memiliki masalah tersebut. Masalah yang dimaksud oleh Kroenke ini sering disebut dengan istilah anomali.
- Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / database, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi).
- Normalisasi adalah suatu proses memperbaiki / membangun dengan model data relasional, dan secara umum lebih tepat dikoneksikan dengan model data logika.

- Menurut E.F. Codd:
  - Normalisasi dipakai untuk membuat struktur tabel (relasi) dalam basis data mengurangi kemubaziran data.
  - Kadang dipakai sebagai perangkat verifikasi terhadap tabel-tabel yang dihasilkan metodologi lain.
  - Memberikan panduan yang sangat membantu bagi pengembang untuk mencegah penciptaan struktur tabel yang kurang atau mengurangi ketidak-efisienan.
- Menurut Kroenke:

  Sebagai proses untuk mengubah suatu relasi yang memiliki masalah tertentu ke dalam dua buah relasi atau lebih yang tidak memiliki masalah tersebut anomali: Proses pada basis data yang memberikan efek samping yang tidak diharapkan.

b. Tabel Relasi

Ditujukan untuk mensingkronisasikan hubungan antar tabel yang bersangkutan untuk menciptakan keselarasan antar tabel.