

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Konsep dasar Sistem Informasi**

Konsep dasar dari Sistem Informasi terbagi atas dua pengertian. Yang pertama adalah sistem, dan yang kedua adalah sistem informasi itu sendiri.

##### **3.1.1 Sistem**

Definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya. (Herlambang, 2005)

##### **3.1.2 Sistem Informasi**

Data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, Informasi adalah data yang telah diolah dan

mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya. (Herlambang, 2005)

### 3.2 Manajemen Aset

Aset adalah sumber daya yang mempunyai manfaat ekonomik masa datang yang cukup pasti atau diperoleh atau dikuasai/dikendalikan oleh suatu entitas akibat transaksi atau kejadian masa lalu.

Aset mempunyai sifat sebagai manfaat ekonomik dan bukan sebagai sumber ekonomik karena manfaat ekonomik tidak membatasi bentuk atau jenis sumber ekonomik yang dapat dimasukkan sebagai aset. Aset pada umumnya terbagi 2 yaitu aset tetap dan aset tidak berwujud.

Aset tetap adalah aset berwujud yang mempunyai masa manfaat lebih dari 12 (dua belas) bulan untuk digunakan dalam kegiatan ekonomi perusahaan. Aset tetap diklasifikasikan berdasarkan kesamaan dalam sifat atau fungsinya dalam aktivitas operasi entitas tanah, peralatan, gedung bangunan, jalan dan sebagainya. Aset tidak berwujud adalah jenis aset yang tidak memiliki wujud fisik. Contoh dari aset ini adalah hak cipta, paten, merek dagang, rahasia dagang.

Siklus manajemen aset mempertimbangkan semua pilihan dan strategi manajemen sebagai bagian dari aset masa pakai, dari perencanaan sampai penghapusan aset. Tujuan adalah untuk mencari biaya terendah dalam jangka panjang (bukan penghematan dalam jangka pendek) ketika membuat keputusan dalam manajemen aset.

Perencanaan aset meliputi konfirmasi tentang pelayanan yang dibutuhkan oleh pelanggan dan memastikan bahwa aset yang diajukan merupakan solusi yang paling efektif untuk memenuhi kebutuhan pelanggan.

Pengadaan aset merupakan peningkatan dari aset dimana pembiayaan dapat menjadi alasan yang diharapkan untuk menyediakan keuntungan diluar tahun pembiayaan.

Pengoperasian aset mempunyai fungsi yang berhubungan dengan kerja, pengendalian aset dan biaya yang berhubungan dengannya yang merupakan komponen penting dalam aset yang dinamis atau berumur pendek. Penghapusan aset adalah pilihan ketika sebuah aset tidak diperlukan lagi, menjadi tidak ekonomis untuk di rawat atau direhabilitasi. Dalam manajemen aset dikenal dengan adanya suatu siklus hidup pengelolaan aset yang biasa dengan *Lifecycle Asset Management*, terdiri dari :

- a. *Asset Planning* (perencanaan aset) meliputi konfirmasi tentang pelayanan yang dibutuhkan oleh pelanggan dan memastikan bahwa aset yang di ajukan merupakan solusi yang apling efektif untuk memenuhi kebutuhan pelanggan.
- b. *Asset creation/acquisition* (pengadaan aset) merupakan pengadaan atau peningkatan dari aset dimana pembiayaan dapat menjadi alasan yang diharapkan untuk menyediakan keuntungan di luar tahun pembiayaan
- c. *Financial management* (manajemen keuangan) merupakan pengetahuan yang berhubungan dengan kepemilikan aset, termasuk pengadaan/akusisi, operasi, maintenance, rehabilitasi, pembaharuan, depresiasi dan pembuangan dan pengambilan keputusan yang mendukung keefektifan biaya yang dikeluarkan.

- d. *Asset operation and maintenance* (perawatan dan pengoperasian aset) mempunyai fungsi yang berhubungan dengan kerja dan pengendalian aset dari hari ke biaya yang berhubungan dengan yang merupakan komponen penting dalam aset yang dinamis atau berumur pendek.
- e. *Asset condition and performance* (kondisi dan kinerja aset) dimana kinerja aset berhubungan dengan pada kemampuan dari aset untuk memenuhi target dari level layanan dan kondisi aset mencerminkan kondisi fisik dari aset.
- f. *Asset rehabilitation/replacement* (rehabilitasi/penggantian aset) adalah upgrade atau penggantian yang cukup signifikan dari sebuah aset atau komponen aset untuk mengendalikan aset kepada kondisi dan kerja yang dibutuhkan.
- g. *Asset disposal/rationalisation* (pembuangan/rasionalisasi aset) adalah pilihan ketika sebuah aset tidak diperlukan lagi, menjadi tidak ekonomis untuk dirawat atau direhabilitasi.
- h. *Asset management review* (review manajemen aset) melibatkan regulasi internal dan audit independen untuk meyakinkan siklus peningkatan aset manajemen yang kontinyu dan untuk mencapai atau pemelihara praktik terbaik bagi perusahaan. (Mitchell, 2006)

### **3.3 Analisa dan Perancangan Sistem**

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan,

hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.

Analisis sistem dilakukan setelah tahap perencanaan sistem dan sebelum tahap perancangan sistem. Langkah-langkah dasar dalam melakukan analisa sistem :

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kinerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisa sistem
4. *Report*, yaitu membuat laporan hasil analisis

Setelah analisis sistem dilakukan, tahap selanjutnya adalah perancangan sistem. Perancangan sistem mempunyai dua tujuan utama, yaitu untuk memenuhi kebutuhan kepada pemakai serta memberikan gambaran yang jelas dan lengkap kepada pemrogram dan ahli teknik lainnya yang terlibat (Jogiyanto, HM, 1990).

### 3.4 System Flow

*System flow* adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang ada (Jogiyanto, 1998).

Terdapat berbagai macam bentuk simbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *decision*, *off-line storage*, *on-page reference*, dan *off-page reference*.

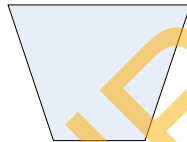
*Terminator* merupakan bentuk simbol yang digunakan sebagai tanda dimulainya jalan proses sistem ataupun tanda akhir dari sebuah pengerjaan suatu sistem.

Bentuk dari *terminator* adalah sebagai berikut:



Gambar 3.1. *Simbol Terminator*

*Manual operation* digunakan untuk menggambarkan sebuah proses kerja yang dilakukan tanpa menggunakan komputer sebagai medianya (menggunakan proses manual). Bentuk simbolnya adalah:



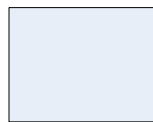
Gambar 3.2. *Simbol Manual Operation*

*Document* merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik. Bentuk dari *document* di gambarkan dalam simbol berikut:



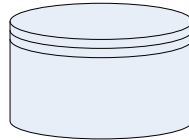
Gambar 3.3. *Simbol Document*

*Process* adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. *Process* disimbolkan dengan gambar:



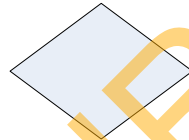
Gambar 3.4. *Simbol Process*

*Database* digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* adalah sebagai berikut:



Gambar 3.5. *Simbol Database*

*Decision* merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar dan salah. Operator logika ini digambarkan sebagai berikut:



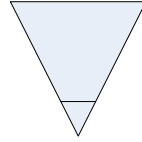
Gambar 3.6. *Simbol Decision*

*Manual input* digunakan untuk melakukan proses *input* ke dalam *database* melalui *keyboard*. *Manual input* digambarkan dengan simbol:



Gambar 3.7. *Simbol Manual Input*

*Off-line storage* merupakan bentuk media penyimpanan yang berbeda dengan *database*, dimana media penyimpanan ini menyimpan dokumen secara manual atau lebih dikenal dengan nama arsip. *Off-line storage* digambarkan dengan simbol:



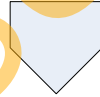
Gambar 3.8. *Simbol Off-Line Storage*

*On-page reference* digunakan sebagai simbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh dalam permasalahan letaknya. Bentuk simbol *On-page reference* adalah sebagai berikut:



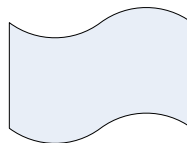
Gambar 3.9. *Simbol On-Page Reference*

*Off-page reference* memiliki sifat yang sedikit berbeda dengan *On-page reference*, karena simbol ini hanya digunakan apabila arus data yang ada dilanjutkan ke halaman yang berbeda. Bentuk simbolnya adalah:



Gambar 3.10. *Simbol Off-Page Reference*

*Paper tape* merupakan sebuah simbol yang umumnya menggantikan bentuk penggambaran jenis pembayaran yang digunakan (misal: uang) dalam transaksi yang ada pada sistem yang dirancang. Bentuk dari *paper tape* adalah dengan simbol:



Gambar 3.11. *Simbol Paper Tape*

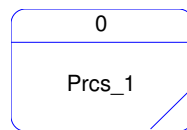


### 3.5 Data Flow Diagram

*Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data tersebut disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

*Data Flow Diagram* merupakan suatu metode pengembangan sistem yang terstruktur (*structured analysis and design*). Penggunaan notasi dalam *data flow diagram* sangat membantu untuk memahami suatu sistem pada semua tingkat kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

Di dalam *data flow diagram* terdapat empat simbol yang digunakan yaitu *process*, *external entity*, *data store*, dan *data flow*. Simbol *process* digunakan untuk melakukan suatu perubahan berdasarkan data yang diinputkan dan menghasilkan data dari perubahan tersebut. Simbol *process* dapat digambarkan sebagai bentuk berikut:



Gambar 3.12. Simbol Process

Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas proses. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa proses tersebut adalah sebuah *context diagram*. Diagram ini merupakan level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Pembuatan *context diagram* dapat dilakukan dengan terlebih dahulu

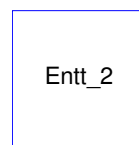
menentukan nama sistemnya, menentukan batasan dari sistem, dan menentukan *terminator* yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

Nomor 1, 2, 3, dan seterusnya menandakan bahwa proses tersebut diartikan sebagai proses level-0 (nol) yang merupakan hasil turunan atau *decompose* dari proses *context diagram*. Proses level-0 membahas sistem secara lebih mendetil, baik dipandang dari segi kegiatan dari sebuah bagian, alur data yang ada, maupun *database* yang digunakan didalamnya. Pembuatannya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem, menentukan alur data yang diterima dan diberikan masing-masing proses daripada sistem sambil memperhatikan konsep keseimbangan (alur data yang masuk atau keluar dari suatu level harus sama dengan alur data yang masuk dan keluar pada level berikutnya), memunculkan *data store* sebagai sumber maupun tujuan data (*optional*), menggambarkan diagram level-0, menghindari perpotongan arus data, dan melakukan pemberian nomor pada proses utama (nomor tidak menunjukkan urutan proses).

Nomor 1.1, 1.2, 2.1, 2.2, dan seterusnya merupakan sebuah proses turunan atau *decompose* dari proses level-0 yang disebut sebagai proses level-1 (satu). Proses level-1 menggambarkan detil kerja dari sebuah bagian dalam sebuah sistem. Penggambarannya dilakukan dengan cara menentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level-0, menentukan apa yang diterima atau diberikan masing-masing sub-proses daripada sistem dan tetap memperhatikan konsep keseimbangan, memunculkan *data store* sebagai sumber maupun tujuan alur data (*optional*), menggambar DFD level-1, dan berusaha

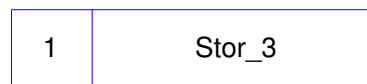
untuk menghindari perpotongan arus data. Hasil turunan akhir disebut sebagai *the lowest level*, dimana hasil akhir ini tergantung dari kompleksitas sistem yang ada.

*External entity* disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku-pelaku sistem yang terkait, dapat berupa orang-orang, organisasi maupun instansi. *External entity* dapat memberikan masukan kepada *process* dan mendapatkan keluaran dari *process*. *External entity* digambarkan dalam bentuk sebagai berikut:



Gambar 3.13. Simbol *External Entity*

*Data store* digunakan sebagai media penyimpanan suatu data yang dapat berupa *file* atau *database*, arsip atau catatan manual, lemari *file*, dan tabel-tabel dalam *database*. Penamaan *data store* harus sesuai dengan bentuk data yang tersimpan pada *data store* tersebut, misalnya tabel pelamar, tabel pendidikan, tabel lulus seleksi, dan lain-lain. *Data store* digambarkan dalam bentuk simbol sebagai berikut:



Gambar 3.14. Simbol *Data Store*

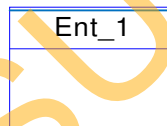
*Data flow* merupakan penghubung antara *external entity* dengan *process* dan *process* dengan *data store*. *Data flow* menunjukkan aliran data dari satu titik ke titik lainnya dengan tanda anak panah mengarah ke tujuan data. Penamaan *data flow* harus menggunakan kata benda, karena di dalam *data flow* mengandung sekumpulan data. *Data flow* digambarkan dengan bentuk simbol sebagai berikut:

Gambar 3.15. Simbol *Data Flow*

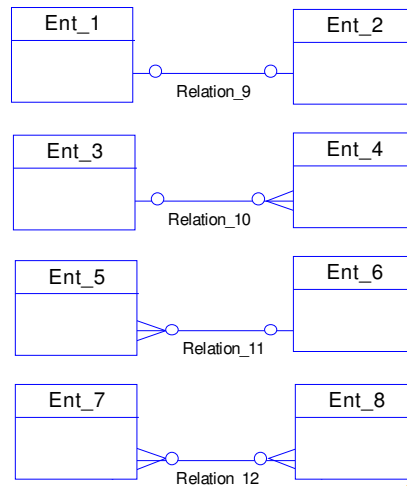
### 3.6 Entity Relationship Diagram

*Entity Relationship Diagram*, atau yang lebih dikenal dengan nama ERD, digunakan untuk mengimplementasikan, menentukan, dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari pemakai. Adapun elemen-elemen yang terdapat pada ERD, adalah sebagai berikut:

1. *Entity* atau entitas, digambarkan dalam bentuk persegi seperti pada gambar 3.16.

Gambar 3.16. Simbol *Entity* atau Entitas

2. *Relation* atau relasi merupakan penghubung antara entitas dengan entitas. Terdapat beberapa jenis relasi yang dapat digunakan, seperti *one-to-one*, *one-to-many*, *many-to-one*, dan *many-to-many*. Bentuk alur relasi secara detail dapat dilihat pada gambar 3.17.



Gambar 3.17. *Simbol Relation of Entity*

### 3.6 Microsoft Visual Basic.Net

Menurut Yuswanto (2006) Visual Basic (VB) 2005 merupakan bahasa pemrograman yang terdapat dalam satu paket aplikasi Visual Studio 2005 merupakan suatu produk Microsoft yang merupakan penerus dari Visual Studio 2003. Budiharto menyebutkan, “Visual Basic 2005 ialah bahasa pemrograman terbaru yang memudahkan programmer VB 6/VB.Net beralih ke VB 2005”. Budiharto juga menyebutkan alasan penting lainnya untuk melakukan migrasi VB 2005, yaitu:

- a. Visual Basic 2005 mengatasi semua masalah yang sulit di sekitar pengembangan aplikasi berbasis Windows dan mengurangi penggunaan aplikasi lainnya serta versi komponen, bahkan mewarisi sifat C++ dan berbau Java.
- b. Visual Basic 2005 memiliki fasilitas penanganan bug yang hebat dan *real time background copiler* yang mengakibatkan developer visual C# dapat mengetahui kesalahan kode yang terjadi secara up-to-date.

- c. *Windows Form designer* memungkinkan developer memperoleh aplikasi desktop dalam waktu yang singkat.
- d. Bagi developer, Visual Basic 2005 menyediakan model pemrograman data akses ActiveX Data Object (ADO) yang sudah dikenal dan diminati, ditambah XML baru yang berbasis Microsoft ADO.Net dengan ADO.Net, developer akan memperoleh akses ke komponen yang lebih powerfull, seperti *control DataSet*.
- e. Visual Basic 2005 menghasilkan web. Menggunakan form web yang baru, anda dapat dengan mudah membangun *thin-client* aplikasi berbasis web yang secara cerdas berjalan di browser dan platform manapun.
- f. Mendukung pembangunan Aplikasi client-server, terdistribusi serta berupa aplikasi yang berbasis Windows serta web.
- g. .Net Framework secara mendasar dibuat untuk dipasangkan dengan Windows 2003 dengan keunggulan memonitor kelalaian dari aplikasi yang sedang berjalan, dan mengisolasi setiap aplikasi yang sedang berjalan dan mengisolasi setiap aplikasi.
- h. Developer dengan berbagai latar belakang dapat dengan segera menguasai .Net karena kemudahan dan kemiripan kode yang ditawarkannya.
- i. Deployment/Penyebaran yang mudah, baik untuk aplikasi windows maupun aplikasi web karena sudah tersedia wizard atau tool secara khusus dengan fasilitas tambahan yang menarik. Tool canggih ini tidak tersedia pada aplikasi sebelumnya bahkan pada bahasa pemrograman lain.
- j. Integrasi dengan system yang sudah ada sangat mudah, Net Framework com memungkinkan Anda erinteraksi dan dengan sistem yang sudah ada

menggunakan XML web Service. Terakhir, Visual Studio Upgrade tool yang tersedia pada Visual Studio.Net dan Java Language Convention Assistant membantu Anda mengkonversi Visual Basic 6 dan Visual J++ agar berjalan pada .Net Framework.

- k. Mendukung lebih dari 20 bahasa pemrograman, Net Framework mendukung integrasi lebih dari 20 bahasa pemrograman yang tidak terbayang sebelumnya. Memungkinkan pengembang memilih bahasa pemrograman yang tepat sesuai latar belakang pemrogramannya.

### **3.8 Microsoft SQL Server 2005**

SQL Server 2005 merupakan produk dari Microsoft dalam bidang *Relational Database Management System* (RDBMS) yang didesain untuk mendukung proses transaksi yang besar. SQL Server 2005 dapat dijalankan pada Windows 2000 Profesional service Pack 4, Windows 2000 service Pack 4, Windows XP Profesional service Pack 2, atau Windows 2003 server service Pack 1. SQL Server 2005 membutuhkan windows installer 3.1 yang dapat diperoleh pada saat instalasi Visual Studio 2005.

SQL Server 2005 terdiri atas tujuh edisi berbeda yang tersedia pada CD yang berbeda, yaitu Standart Edition, Enterprise Edition, Personal Edition, Developer Edition, Windows CE Edition, Evaluation Edition, Evaluation Edition dan Microsoft Desktop Engine (MDE). SQL Server 2005 mempunyai fasilitas tambahan yang membuat software tersebut memiliki kemampuan penuh dalam e-commerce. SQL Server 2005 secara otomatis akan menginstall enam database utama, yaitu master, model, tempdb, pubs, Northwind dan msdb.