BAB III

LANDASAN TEORI

Untuk melakukan perancangan sistem dan pembuatan aplikasi maka diperlukan adanya suatu teori penunjang yang mendasari tentang pemahaman dan konsep sebuah sistem. Adapun pemahaman tentang sistem tersebut memiliki teori-teori penunjang sebagai berikut :

3.1 Konsep Dasar Sistem

Dalam perancangan sistem terlebih dahulu harus mengerti sub sistem. Sub sistem yaitu serangkaian kegiatan yang dapat ditentukan identitasnya, yang berhubungan dengan suatu sistem. Menurut Garvin (1974), sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsepsi yang saling bergantungan. Sedangkan menurut The Liang Gie (1976), sistem adalah suatu kebulatan dari bagian-bagian atau unsur-unsur yang saling berhubungan menurut suatu pengaturan yang tertib guna mencapai maksud tertentu. Dari berbagai pendapat dikemukakan semuanya mempunyai maksud bahwa sistem bertanggung jawab memproses masukan (*input*) dan kemudian menghasilkan keluaran (*output*).

3.1.1 Sistem Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian nyata sering terjadi adalah perubahan dari suatu nilai yang disebut transaksi.

Sistem adalah sekumpulan komponen yang dirangkai untuk suatu tujuan tertentu. Sistem merupakan buatan manusia yang terdiri dari himpunan yang terintegrasi dari komponen-komponen manual dan komponen-komponen terkomputerisasi yang bertujuan untuk mengumpulkan, menyimpan, memproses data dan menghasilkan informasi yang digunakan untuk menentukan keputusan bagi *user*.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi mempunyai komponen-komponen yaitu :

1. Blok masukan, *input* mewakili data yang masuk ke dalam sistem informasi.
2. Blok manual, terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input*.
3. Blok teknologi, untuk menerima *input*, menjalankan model, menyimpan data, dan mengakses data.
4. Blok basis data, kumpulan dari data yang saling berhubungan satu dengan yang lain , tersimpan di perangkat lunak.
5. Blok kendali, untuk mencegah hal-hal yang tidak diinginkan yang dapat merusak suatu sistem.

3.1.2 Unsur-Unsur Sistem Informasi

Sistem informasi, yang memiliki tiga kegiatan utama, yaitu menerima data sebagai masukan kemudian memprosesnya dengan perhitungan dan penggabungan unsur data dan akhirnya memperoleh informasi sebagai keluaran (*output*). Secara sederhana sebuah sistem informasi menerima dan memproses data dan kemudian mengubahnya menjadi informasi.

3.2 System Flow

System Flow adalah suatu bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagian ini menjelaskan urutan prosedur yang ada di dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pada fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub sistem.

System flow digambarkan dengan menggunakan symbol-simbol yang tampak seperti gambar 3.1 :

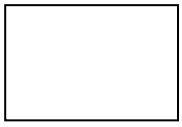
(a) Simbol dokumen

Menunjukkan dokumen *input* dan *output*, baik untuk proses manual mekanik ataupun komputer

(b) Simbol kegiatan manual

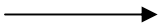
Menunjukkan pekerjaan manual

(c) Simbol proses



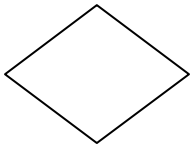
Menunjukkan kegiatan proses dari operasi program komputer

(d) Simbol garis alir



Menunjukkan arus dari proses

(e) Simbol *decision* / keputusan



Menunjukkan keputusan dari suatu proses

Gambar 3.1 Simbol-Simbol dalam System Flow

3.3 Data Flow Diagram

Model sistem logika dari sistem informasi menjelaskan kepada *user* bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. *Logical model* dapat digambarkan dengan menggunakan diagram arus data (*data flow diagram* / DFD). Simbol yang digunakan di DFD dalam buku ini menggunakan simbol dari Gane & Sarson terdiri dari 4 simbol yaitu :

3.3.1 External Entity (Kesatuan Luar) atau Boundary (Batas Sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

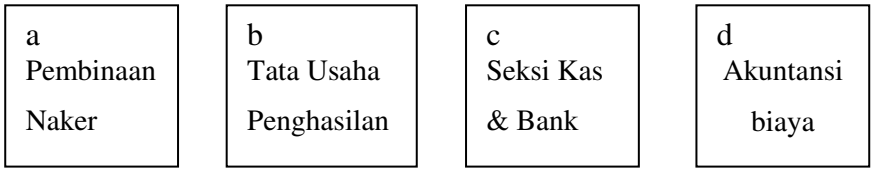
Kesatuan luar ini dapat berupa salah satu dari berikut ini :

1. Suatu kantor, departemen atau divisi dalam perusahaan, tetapi di luar sistem yang sedang dikembangkan.
2. Orang atau sekelompok orang di organisasi, tetapi di luar sistem yang sedang dikembangkan.
3. Suatu organisasi atau orang yang berada di luar organisasi, misalnya langganan, pemasok.
4. Sistem informasi yang lain di luar sistem yang sedang dikembangkan.
5. Sumber asli dari suatu transaksi.
6. Penerima akhir dari suatu laporan yang dihasilkan oleh sistem.

Suatu kesatuan luar dapat disimbolkan dengan suatu notasi kotak seperti gambar di 3.2

Gambar 3.2 Notasi Kesatuan Luar di DFD

Kesatuan luar dapat diberi identifikasi dengan huruf kecil di ujung kiri atas seperti gambar 3.3



Gambar 3.3 Identifikasi Notasi Kesatuan Luar

3.3.2 Data Flow (Arus Data)

Arus data (*data flow*) di *Data Flow Diagram* digambarkan dengan simbol sebuah panah. Arus data ini mengalir diantara proses, simpanan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem, dan dapat berbentuk sesuatu sebagai berikut :

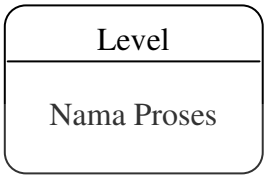
1. Formulir atau dokumen yang digunakan oleh sebuah perusahaan.
2. Laporan tercetak yang dihasilkan oleh sistem.
3. Tampilan atau *output* di layar komputer yang dihasilkan oleh sistem.
4. Masukan untuk komputer.
5. Komunikasi.
6. Surat-surat atau memo.
7. Data yang dibaca atau direkamkan ke suatu file.
8. Suatu isian yang dicatat pada buku agenda.
9. Transmisi data dari suatu komputer ke komputer yang lain.

Arus data sebaiknya diberi nama yang jelas dan mempunyai arti. Nama dari arus data dituliskan di samping garis panahnya seperti pada gambar 3.4.

Gambar 3.4 Arus Data berupa Data Barang

3.3.3 Process (Proses)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses, dan selanjutnya akan dihasilkan arus data yang nantinya juga keluar dari proses tersebut. Suatu proses dapat ditunjukkan dengan simbol empat persegi panjang tegak dengan sudut-sudut yang tumpul seperti tampak pada gambar 3.5.



Gambar 3.5 Notasi Proses di DFD

Setiap proses harus diberi penjelasan yang lengkap meliputi keterangan seperti berikut ini :

1. Identifikasi proses

   Identifikasi ini umumnya berupa suatu angka yang menunjukkan nomor acuan dari proses dan ditulis pada bagian atas di simbol proses.

2. Nama proses

   Nama proses menunjukkan apa yang dikerjakan oleh proses tersebut. Nama dari proses harus jelas dan lengkap yang nantinya akan menggambarkan kegiatan prosesnya. Nama dari proses biasanya diletakkan di bawah identifikasi proses dan berada pada simbol proses.

3.3.4 Data Store/Database (Simpanan Data)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa sebagai berikut ini :

1. Suatu file atau *database* di sistem komputer.
2. Suatu arsip atau catatan manual.
3. Suatu tabel acuan manual.

Simpanan data di DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup di salah satu ujungnya. Nama dari *database* menunjukkan nama dari filenya, misalnya file data *customer*, data iklan dan lain sebagainya. Penjelasan dapat dilihat seperti pada gambar 3.6.

Gambar 3.6 simbol Database di DFD

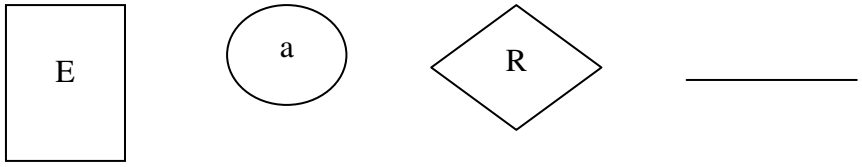
3.4 Entity Relational Diagram (ER- Diagram)

Struktur logika dari basis data dapat diekspresikan secara grafik dengan diagram E-R, yang terdiri dari komponen-komponen :

1. Persegi panjang, yaitu menyatakan himpunan entitas
2. Lingkaran/Ellips, yaitu menyatakan atribut
3. Belah ketupat, yaitu menyatakan himpunan relasi
4. Garis, yaitu sebagai penghubung antara himpunan relasi dengan himpunan entitas, dan himpunan entitas dengan atributnya

5. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka.

   Gambar 3.7. di bawah ini menunjukkan komponen-komponen diagram E-R.



(1) Entitas E | (2) Atribut a | (3) Relasi R | (4) Link

Gambar 3.7 Simbol Komponen E-R Diagram

Model E-R dibentuk dari dua komponen utama, yaitu entitas (*entity*) dan relasi (*relation*). Kedua komponen ini dideskripsikan lebih jauh melalui sejumlah atribut/properti.

1. Entitas dan himpunan entitas

   Entitas adalah obyek yang ada dan dapat dibedakan dengan obyek yang lain. Sedangkan himpunan entitas adalah sebuah himpunan entitas dan mempunyai tipe yang sama.

2. Atribut/Properti

   Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (properti) dari entitas tersebut. Penentuan atau pemilihan atribut yang *relevan* bagi sebuah entitas .

3. Relasi dan himpunan relasi

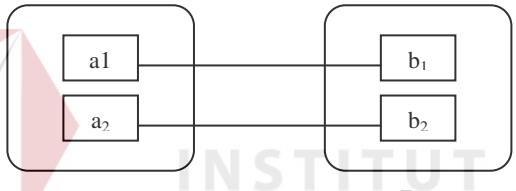
   Relasi menunjukkan hubungan diantara sejumlah entitas yang berbeda.

4. Kardinalitas

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa :

a. One-to-One.

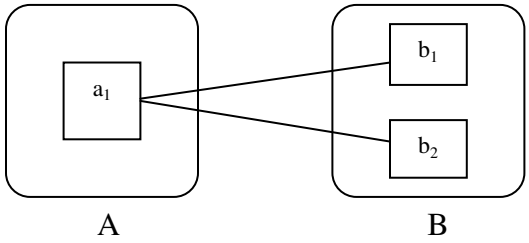
Sebuah entitas di A dihubungkan dengan satu entitas di B dan sebuah entitas di B dihubungkan dengan satu entitas di A. Gambar 3.8 menunjukkan relasi *One-to-One*.



Gambar 3.8 Relasi *One-to-One*

b. One-to-Many.

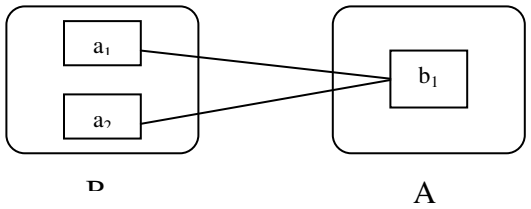
Sebuah entitas di A dihubungkan dengan sejumlah entitas di B dan sebuah entitas di B dihubungkan dengan satu entitas di A. Gambar 3.9 menunjukkan relasi *One-to-Many*.



Gambar 3.9 Relasi *One-to-Many*

c. Many-to-One.

Sebuah entitas di A dihubungkan dengan sejumlah entitas di B dan sebuah entitas di B dihubungkan dengan sejumlah entitas di A. Gambar 3.10 menunjukkan relasi *Many-to-One*.

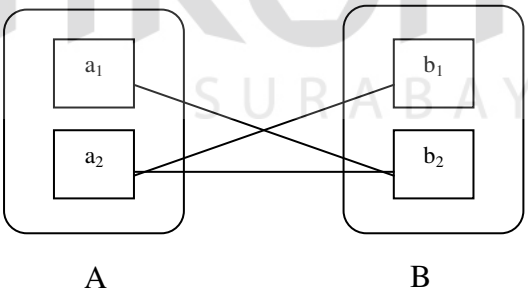


Gambar 3.10 Relasi *Many-to-One*

d. Many-to-Many.

Sebuah entitas di A dihubungkan dengan sejumlah entitas di B dan sebuah entitas di B dihubungkan dengan sejumlah entitas di A. Gambar 3.11 menunjukkan relasi *Many-to-Many.*

.



Gambar 3.11 Relasi *Many-to-Many*

3.5 *Visual Basic .NET*

Visual Basic versi sebelumnya yaitu *Visual Basic* 6 diluncurkan oleh *Microsoft* pada tahun 1998. Kemudian setelah beberapa tahun, *Microsoft* memaparkan pengembangan *Microsoft .NET* dalam PDC *(Professional Developers Conference)* di Orlando, Florida, Amerika Serikat pada bulan Juli 2000.

Visual Basic .NET (selanjutnya disebut sebagai VB.NET) bukan merupakan *Visual Basic* 6+1, banyak hal yang tidak bisa dikerjakan oleh *Visual Basic* 6 dapat dikerjakan di *Visual Basic .NET*. Berikut adalah kelebihan yang dimiliki oleh *Visual Basic .NET* :

1. Menyederhanakan *Deployment*.

   VB.NET mengatasi masalah seputar *deployment* dari aplikasi berbasis *Windows* yaitu "DLL *Hell*" dan registrasi COM (*Component Object Model*). Secara berdampingan *versioning* (pengaturan versi komponen) mencegah tertindihnya dan terkorupsinya komponen dan aplikasi. *Deployment* secara XCOPY memungkinkan pengembang menginstal aplikasi berbasis *windows* ke mesin *client* cukup dengan cara menyalin file ke suatu direktori. *Deployment* tidak perlu melakukan registrasi dan tidak perlu GUID serta prosedur instalasi.

   Fitur *auto-downloading,* akan mempermudah *deployment* aplikasi berbasis *windows* melalui Internet karena aplikasi dapat diinstal dan dijalankan dengan *pointing/menunjuk* dari *browser web* ke suatu URL (*Universal Resources Locator*). URL adalah alamat lengkap yang menunjuk ke halaman tertentu atau file tertentu dalam suatu *browser*.

2. Menyederhanakan Pengembangan Perangkat Lunak.

   VB.NET memiliki fitur *compiler* yang bekerja secara *background real-time* dan daftar *task* untuk penanganan *bug* program sehingga pengembang dapat langsung memperbaiki kesalahan kode yang terjadi.

   Bahasa pemrograman *Visual Basic .NET* sekarang diperkaya, seperti adanya kata kunci *Option Strict On* untuk menampilkan kesalahan sintaks jika Anda membuat konversi tipe data secara eksplisit yang bisa menyebabkan hilangnya data. Juga adanya penanganan eksepsi (kejadian yang tidak diharapkan yang muncul saat kode dieksekusi) terstruktur yang baru menggunakan *Try... Catch.. Finally*.

   Desainer form *windows* yang baru memungkinkan pengembang membuat aplikasi *desktop* dalam waktu yang singkat. Fitur baru seperti kontrol *docking* (menambatkan) dan *anchorizing* (jangkar) dapat digunakan mengatasi masalah resize ukuran jendela. Editor menu yang bersifat WYSIWYG *(What You See Is What You Get)* akan menyederhanakan langkah-langkah dalam mendesain menu. Fitur *editor tab order* dapat memudahkan mengorganisasikan kontrol.

   Pada pemrograman database disediakan teknologi ADO.NET (*ActiveX Data Objects* untuk *NET Framework*) yang merupakan kumpulan *class* dan berisi komponen untuk melakukan koneksi, akses dan manipulasi *database*. Dalam ADO.NET terdapat provider data *SQL Server* dan OLE DB. Pada VB.NET versi 2003 provider datanya ditambah dengan *Oracle* dan ODBC. Ketika sedang menulis kode untuk akses data, *Intellisense* secara otomatis akan membuat lengkap

suatu pernyataan kode. Adanya fitur *Data Form Wizard*, akan mempercepat pembuatan aplikasi *database.*

3. Mendukung penuh *Object Oriented Programming*

   Dalam *Visual Basic .NET*, Anda dapat membuat kode *class* yang menggunakan secara penuh konstruksi berbasis obyek. *Class-class* tersebut *reusable* dan dapat digunakan kembali. *Visual Basic .NET* memiliki fitur bahasa pemrograman berorientasi objek termasuk implementasinya secara penuh: *inheritance*, *encapsulation,* dan *polymorphism.*

3.6 *Database Management System* (DBMS)

Database adalah sekumpulan data/informasi yang tersimpan secara teratur berdasarkan kriteria/kelompok tertentu, dimana kriteria tersebut saling berhubungan satu sama lain. Sedangkan *Database Management System (DBMS)* adalah kumpulan program perangkat lunak (*software*) yang memperbolehkan *user* untuk membuat dan memelihara *database*. Semua *DBMS* tersebut disatukan oleh *Relational Database Management System (RDBMS).*

3.6.1 *Structured Query Language* (SQL)

SQL pertama kali didefinisikan oleh Donald D. Chamberlain dan Raymond F. Boyce pada tahun 1970 dan dinamakan *Structured English Query Language* (SEQUEL). Pada perkembangannya SEQUEL diubah namanya menjadi SQL dan pada tahun 1986 oleh American Nasional Standart Institude (ANSI) didefinisikan sebagai SQL-92 dan distandarisasi oleh International Standard Organization sebagai ISO/IEC 9057: 1992, "*Database Language* SQL". SQL merupakan bahasa standar

untuk melakukan proses *query* terhadap basis data (http://en.wikipedia.org/wiki/SQL, diakses 29 Maret 2012 21:34).

3.6.2 *Microsoft SQL Server*

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional(RDBMS) yang diproduksi oleh *Microsoft*. *Query* utamanya adalah *Transact-SQL* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar.

Microsoft SQL Server dan *Sybase*/ASE dapat berkomunikasi lewat jaringan ataupun *web service* dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, *Microsoft SQL Server* juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai *driver* JDBC untuk bahasa pemrograman *Java*. Fitur yang lain dari *SQL Server* ini adalah kemampuannya untuk membuat basis data *mirroring* dan *clustering*. (http://id.wikipedia.org/wiki/Microsoft_SQL_Server diakses 29 Maret 2012 22:02).

3.7 Short Message Service

Short Message Service (SMS) atau layanan pesan singkat mempunyai sejarah tersendiri sebagai media layanan yang paling meledak abad ini. Awalnya SMS berfungsi untuk memberikan layanan pengiriman pesan teks singkat antar perangkat *mobile phone* (telepon genggam/ *handphone*). SMS sebetulnya hanya layanan tambahan terhadap dua layanan utama (layanan *voice* dan *switched data*)

dalam sistem jaringan komunikasi GSM. GSM (*Global System for Mobile Communications*) adalah perkumpulan penyedia perangkat komunikasi Eropa yang menyediakan standardisasi perangkat telepon genggam / telepon bergerak di Eropa. Namun karena keberhasilan SMS yang tidak terduga, dengan pelanggan yang menggunakannya, menjadikan SMS sebagai bagian *integral* dari layanan sistem *standart-standart* komunikasi lain, seperti CDMA, UMTS, bahkan jaringan telepon rumah (*fixed phone*) bahkan mulai mengadopsi teknologi yang sebetulnya sangat sederhana ini. Aplikasi ini hanya terbatas pada pengiriman dan penerimaan data berupa teks dengan panjang pesan antara 120-160 huruf bahkan ada yang sampai 765 huruf", (Baharuddin, 2008).

3.8 SMS Gateway

Salah satu mode komunikasi yang handal saat ini adalah pesan pendek / SMS. Implikasinya, salah satu model komunikasi data yang bisa dipakai adalah SMS. Artinya, SMS tersebut harus bisa melakukan transaksi dengan *database*. Untuk itu perlu dibangun sebuah sistem yang disebut sebagai SMS *Gateway*. Pada prinsipnya, SMS *Gateway* adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan guna mendistribusikan pesan-pesan yang di-*generate* lewat sistem informasi melalui media SMS yang ditangani oleh jaringan seluler. Secara khusus, sistem ini akan memiliki fungsi-fungsi sebagai berikut:

1. *Message Management*

   Pengaturan pesan yang meliputi manajemen waktu pengiriman pesan serta tujuan penerima pesan.

2. Korelasi

   Berfungsi untuk melakukan korelasi data untuk menghasilkan data terbaru hasil korelasi dengan *database*.

Dalam membuat aplikasi dengan menggunakan teknologi SMS *Gateway* pada *Visual Basic* maka dibutuhkan komponen yaitu *GSMComm.*

3.9 GSMComm

GSMComm merupakan komponen .Net yang digunakan untuk SMS *Gateway* yang mendukung dengan perangkat GSM yang dapat dikembangkan menggunakan *Windows Software Developer* (*VB.Net, C#.Net, Visual C++* serta *ASP.Net*).

GSMComm dapat di *install* di semua sistem operasi yang berbasis windows dan bersifat *free license.*