

BAB III

LANDASAN TEORI

3.1 Konsep Penjualan

3.1.1 Pengertian Penjualan

Penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana- rencana strategis yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembeli guna mendapatkan penjualan yang menghasilkan laba (Marwan A, 1986). Penjualan merupakan sumber hidup suatu perusahaan, karena dari perusahaan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan untuk mengetahui daya tarik mereka sehingga dapat mengetahui hasil produk yang dihasilkan. Menurut Winardi (1982), penjualan adalah suatu transfer hak atas benda-benda. Dari penjelasan tersebut dalam memindahkan atau mentransfer barang dan jasa diperlukan orang-orang yang bekerja di bidang penjualan seperti pelaksanaan dagang, agen, wakil pelayanan, dan wakil pemasaran.

3.1.2 Jenis Penjualan

Penjualan jika diidentifikasi dari Perusahaannya dibedakan menjadi beberapa jenis (Martin, S dan Colleran, G 2006) :

1. Penjualan langsung yaitu penjualan dengan mengambil barang dari supplier dan langsung dikirim ke customer.
2. Penjualan stok gudang yaitu penjualan barang dari stok yang telah tersedia di gudang.

3. Penjualan kombinasi yaitu penjualan dengan mengambil barang sebagian dari supplier dan sebagian dari stok yang tersedia di gudang.

3.2 Analisa dan Perancangan Sistem

Menurut Kendall (2003:7), Analisis dan Perancangan Sistem berupaya menganalisis input data atau aliran data secara sistematis, memproses atau mentransformasikan data, menyimpan data, dan menghasilkan *output* informasi dalam konteks bisnis khusus. Selanjutnya, analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang bisa dicapai melalui penggunaan sistem informasi terkomputerisasi.

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini juga akan menyebabkan kesalahan di tahap selanjutnya. Dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut:


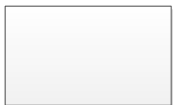






1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

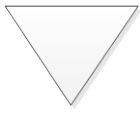
Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut, tahap ini disebut desain sistem.

3.2.1 Bagan Alir Dokumen

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan dokumen gambaran arus data dengan menggunakan simbol seperti yang dijelaskan pada tabel 3.1.

Tabel 3.1 *Simbol*

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
1.		Dokumen	Untuk menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer.
2.		Proses Komputerisasi	Menunjukkan kegiatan dari operasi program komputer.
3.		<i>Database</i>	Untuk menyimpan data.
4.		Penghubung	Menunjukkan hubungan di halaman yang sama.
5.		Penghubung Halaman Lain	Menunjukkan hubungan di halaman lain.
6.		<i>Terminator</i>	Menandakan awal/akhir dari suatu sistem.
7.		<i>Decision</i>	Menggambarkan logika keputusan dengan nilai <i>true</i> atau <i>false</i> .
8.		Kegiatan Manual	Untuk menunjukkan pekerjaan yang dilakukan secara manual.

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
9.		Simpanan <i>Offline</i>	Untuk menunjukkan file non-komputer yang diarsip urut angka.

3.2.2 Entity Realtionship Diagram

Entity Relationship Diagram adalah suatu bentuk perencanaan *database* secara konsep fisik yang nantinya akan dipakai sebagai kerangka kerja dan pedoman dari struktur penyimpanan data. ERD digunakan untuk menggambarkan model hubungan data dalam sistem, dimana didalamnya terdapat hubungan entitas beserta atribut relasinya dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan data. ERD memiliki beberapa jenis model yaitu :

Tabel 3.2 Jenis ERD

No.	Jenis ERD	Keterangan
1.	<i>Conceptual Data Model (CDM)</i>	Merupakan model <i>universal</i> dan dapat menggambarkan semua struktur <i>logic database</i> (DBMS), dan tidak bergantung dari <i>software</i> atau pertimbangan struktur <i>data storage</i> . Sebuah CDM dapat diubah langsung menjadi PDM.
2.	<i>Physical Data Model (PDM)</i>	Merupakan model ERD yang mengacu pada pemilihan <i>software</i> DBMS yang spesifik. Hal ini seringkali berbeda secara signifikan dikarenakan oleh struktur tipe <i>database</i> yang bervariasi, dari model <i>schema</i> , tipe data penyimpanan dsb.

3.2.2.1 Jenis Objek ERD

1. *Entity*

Sesuatu yang ada dan terdefiniskan bisa berupa nyata maupun abstrak yang dapat dibedakan satu dengan yang lainnya dan adanya hubungan saling ketergantungan. Ada 2 macam tipe *entity*, yaitu :

a. *Strong Entity*

Strong Entity merupakan tipe *entity* yang mempunyai *key attribute* untuk setiap individu yang ada di dalamnya.

b. *Weak Entity*

Weak Entity merupakan *entity* yang tidak memiliki *key attribute*, oleh karena itu *weak entity* harus dihubungkan dengan *strong entity* untuk menggunakan atribut kunci secara bersama-sama.

2. *Attribute*

Setiap *entity* memiliki beberapa *attribute*, yang merupakan ciri atau karakteristik dari *entity* tersebut. *Attribute* sering disebut juga data elemen atau *data field*.

3. *Key*

Beberapa elemen data memiliki sifat, dengan mengetahui nilai yang telah diberikan oleh sebagian elemen data dari *entity* tertentu, dapat diidentifikasi nilai-nilai yang terkandung dalam elemen-elemen data lain ada *entity* yang sama. Elemen penentu tersebut adalah sebagai elemen data kunci (*key*).

4. *Relationship*

Relationship menggambarkan hubungan yang terjadi antar *entity* yang mewujudkan pemetaan antar *entity*. Bentuk *relationship* yaitu :

a. *One to One Relationship*

Hubungan satu *entity* dengan satu *entity* yang lain.

b. *One to Many Relationship*

Hubungan antar *entity* satu dengan *entity* yang lainnya adalah satu berbanding banyak.

3.2.3 Data Flow Diagram (DFD)

Data Flow Diagram atau sering juga disebut dengan *Bubble Chart* atau diagram, model proses, diagram alur kerja atau model fungsi adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alir data baik secara manual maupun komputerisasi. DFD merupakan alat pembuat model yang sering digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluaran.

Untuk memudahkan proses pembacaan DFD, maka penggambaran DFD disusun berdasarkan tingkatan atau level dari atas ke bawah, yaitu:

1. *Context Diagram*

Merupakan diagram paling atas dan pembahasannya berupa global yang terdiri dari suatu proses dan menggambarkan ruang lingkup proses. Hal yang digambarkan dalam diagram konteks adalah hubungan terminator dengan sistem dan juga sistem dalam suatu proses. Sedangkan hal yang digambarkan dalam *Context Diagram* adalah hubungan antar *terminator* dan *data source*.

2. *Diagram level 0*

Merupakan diagram yang berada diantara diagram konteks dan diagram detail serta menggambarkan proses utama dari *DFD*. Hal yang digambarkan dalam diagram level 0 adalah proses utama dari sistem serta hubungan *entity*, proses, alur data dan *data source*.

3. Diagram Detail (Primitif)

Merupakan penguraian dalam proses yang ada dalam diagram zero.

Diagram yang paling rendah dan tidak dapat diuraikan lagi.

Data Flow Diagram (DFD) memiliki empat komponen, yaitu:

a. Terminator

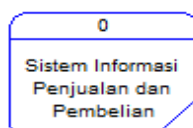
Terminator atau *External Entity* atau kesatuan luar yang mewakili entitas *external* yang berkomunikasi dengan sistem yang dikembangkan. *Terminator* merupakan kesatuan di lingkungan sistem yang dapat berupa orang atau sistem yang berada di lingkungan luar sistem yang memberikan inputan maupun yang menerima output dari sistem serta berupa bagian atau divisi diluar sistem yang berkomunikasi dengan sistem. *Terminator* ini sering juga disebut entitas (*external*).

b. Proses

Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi atau empat persegi panjang tegak dengan sudut tumpul. Proses diberi nama untuk menerangkan proses atau kegiatan apa yang sedang atau kegiatan yang dilaksanakan. Dan setiap proses harus diberi penjelasan yang lengkap sebagai berikut:

1. Identifikasi Proses

Umumnya berupa angka yang menunjukkan nomor dari proses dan



ditulis pada bagian atas simbol.

Gambar 3.1 Simbol Proses

2. Nama Proses

Menunjukkan apa yang sedang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya. Nama proses diletakkan dibawah identifikasi proses.

3. *Data Store*

Data Store digunakan sebagai sarana untuk pengumpulan data. *Data store* disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada satu ujungnya atau dua garis horizontal. Suatu nama perlu diberikan pada *data store* menunjukkan nama filenya. *Data Store* biasanya berkaitan dengan penyimpanan file atau *database* yang dilakukan secara terkomputerisasi. *Data store* dihubungkan dengan alur data hanya pada komponen proses pengertiannya sebagai berikut:

- a. Alur data dari *store* yang berarti sebagai pengaksesan data untuk suatu proses.
- b. Alur data ke proses berarti meng-*update* data seperti menambah data, mengurangi data maupun mengubah data.

4. Alur Data

Alur data dapat digambarkan dengan anak panah yang menuju ke dalam proses maupun ke luar proses. Alur data digunakan untuk

menerangkan perpindahan data atau informasi dari suatu bagian ke bagian lainnya.

5. Syarat-syarat pembuatan sebuah DFD adalah:

- a. Pemberian nama untuk setiap komponen DFD.
- b. Pemberian nomor pada proses DFD.
- c. Menghindari pembuatan DFD yang rumit.
- d. Memastikan DFD dibangun secara konsisten.

a. Perangkat Lunak

i. Visual Basic .NET

Visual Basic (VB) 2005 merupakan bahasa pemrograman yang terdapat dalam satu paket aplikasi Visual Studio 2005. Visual Studio 2005 merupakan suatu produk Microsoft yang merupakan penerus dari Visual Studio 2003. Budiharto(2006:1) menyebutkan, “Visual Basic 2005 ialah bahasa pemrograman terbaru yang memudahkan programmer VB 6/VB.Net beralih ke VB 2005”. Budiharto(2006:3-4) 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 compiler* 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 sistem yang sudah ada sangat mudah, Net Framework com memungkinkan Anda berinteraksi 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.

ii. 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 XP Profesional service Pack 2*, atau *Windows 2003 server service Pack 1* (Budiharto, 2006:21). Budiharto(2006:21) juga menyebutkan 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 Standard Edition, Enterprise Edition, Personal Edition, Developer Edition, Windows CE 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.