

BAB 2

LANDASAN TEORI

2.1 Penjualan Online

Era perkembangan teknologi yang semakin pesat dewasa ini membuat persaingan bisnis dalam bidang apapun menjadi lebih variatif. Banyak metode bisnis yang dilakukan oleh sebuah perusahaan untuk menarik konsumen sebanyak-banyaknya (Kotler,2004,p3). Mulai dari memberikan bonus, memasang iklan, membagikan brosur dan penjualan produk lewat media elektronik seperti internet.

Persaingan bisnis yang semakin sengit ini membuat perusahaan besar maupun perusahaan kecil saling bersaing untuk meningkatkan mutu dan teknik penjualannya. Memasuki era globalisasi ini banyak perusahaan yang menggunakan jasa penjualan *on-line* untuk membantu memasarkan produknya sekaligus memberikan kemudahan kepada konsumennya dalam memilih dan memesan produk yang mereka inginkan.

Dengan menggunakan fasilitas penjualan *on-line* ini, perusahaan berharap konsumen dapat lebih mudah memilih dan memesan produk mana yang mereka inginkan. Dengan fasilitas ini, penjelasan mengenai produk-produk yang dijual menjadi lebih detail sehingga konsumen benar-benar yakin akan apa yang diinginkannya. Selain itu, konsumen dapat melakukan penjualan dengan mudah yakni tanpa beranjak dari tempatnya mengakses web tersebut.

Dewasa ini, penjualan lewat media internet merupakan salah satu strategi perusahaan dalam melayani konsumennya. Diharapkan dengan disediakannya fasilitas ini, jangkauan penjualan dari perusahaan itu sendiri menjadi lebih luas dan konsumen

memperoleh informasi lebih lengkap sebelum memutuskan untuk memesan suatu produk barang.

2.2 Sistem Penjualan

Pengertian sistem penjualan akan dapat lebih dimengerti jika sebelumnya bagian istilah dari sistem informasi didefinisikan terlebih dahulu satu-persatu.

2.2.1 Pengertian Sistem

Ada beberapa macam definisi mengenai sistem, yaitu:

1. McLeod (2001, p11) mengatakan sistem merupakan sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Tanpa memperhatikan asal usulnya, semua sistem memiliki beberapa elemen yang sama.
2. Menurut Pressman (2001, p246), sistem adalah serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan dimana sebelumnya melalui sebuah pemrosesan informasi.

2.2.2 Pengertian Penjualan

Penjualan adalah kegiatan perusahaan untuk mengalihkan kepemilikan, atas barang dan jasa yang telah disediakan untuk dijual kepada pelanggan (Soemarso, 1995, p334). Penjualan antara lain terjadi dari transaksi, penjualan barang atau jasa baik secara tunai maupun kredit (Mulyadi, 2001, p204). Menurut Kotler et.al. (2004, p28), konsep penjualan berpendapat bahwa para konsumen, jika dibiarkan sendiri biasanya cenderung tidak akan membeli produk-produk organisasi tersebut. Oleh karena itu, organisasi harus melakukan kegiatan penjualan yang agresif dan usaha promosi yang gencar.

2.2.3 Pengertian Sistem Penjualan

Menurut Lewison et.al. (1999, p4) penjualan merupakan suatu kegiatan untuk menjual barang-barang atau memberikan pelayanan kepada konsumen akhir.

Berdasar argumen tersebut, maka sistem penjualan dapat diartikan sebagai suatu sistem yang terdiri dari prosedur pencatatan penjualan dan proses pembuatan pernyataan penjualan. Dalam transaksi penjualan terdapat prosedur proses penjualan dengan umumnya dimulai dari penerimaan order dari pelanggan, pengecekan stok, penetapan harga jual hingga penerbitan faktur penjualan yang berdasarkan permintaan order pelanggan.

2.3 Rekayasa Piranti Lunak

Perangkat lunak telah menjadi elemen kunci dari evolusi *computer based-system* dan *computer product*. Menurut Pressman (2001, p20) rekayasa perangkat lunak adalah aplikasi dari metode pendekatan yang dapat dikuantifikasi, disiplin dan sistematis untuk pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dari rekayasa perangkat lunak itu sendiri.

Pada subbab ini akan dibahas mengenai definisi dari *rekayasa piranti lunak*, model pengembangan rekayasa piranti lunak beserta tahap-tahapnya. Pada kasus ini, penulis menggunakan model pengembangan *Waterfall*.

2.3.1 Pengertian Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak adalah aplikasi dari sebuah pendekatan kuantifiabel, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak (Pressman, 1997, p28).

Perangkat lunak atau yang sering disebut *software* itu sendiri adalah perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan. Selain itu ada juga pengertian lain dari perangkat lunak adalah struktur data yang memungkinkan program memanipulasi informasi atau dokumen yang menggambarkan operasi dan kegunaan program (Pressman, 1997, p10).

2.3.2 Paradigma Rekayasa Perangkat Lunak

Paradigma rekayasa perangkat lunak mencakup metode, tools, dan prosedur. Metode rekayasa perangkat lunak memberikan teknik untuk membangun perangkat lunak. Metode-metode itu menyangkut serangkaian tugas yang luas yang menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan.

Tool-tool rekayasa perangkat lunak memberikan topangan yang otomatis ataupun semi otomatis pada proses dan metode yang ada. Prosedur rekayasa perangkat lunak menerapkan metode dan *tools* sekaligus atau pendefinisian urutan aktifitas metode yang diterapkan dan apa yang diinginkan.

2.3.3 Model Proses Perangkat Lunak

Untuk menyelesaikan masalah aktual di dalam sebuah seting industri, rekayasa perangkat lunak atau tim perekayasa harus menggabungkan strategi pengembangan yang melingkupi lapisan proses, metode, dan alat-alat bantu. Strategi ini seting diacukan sebagai model proses. Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat bantu yang akan dipakai dan kontrol serta penyampaian yang dibutuhkan.

Model yang digunakan dalam pembahasan skripsi ini adalah model Waterfall. Model ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial / bertahap yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

Model ini melingkupi aktivitas-aktivitas sebagai berikut :

1. Rekayasa dan pemodelan sistem/informasi

Karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak, manusia, dan database.

2. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perancang perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antar muka yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural.

4. Generasi kode

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional mengarahkan pengujian untuk menemukan kesalahan–kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan–kesalahan ditentukan karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan–perubahan di dalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasi lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.4 Delapan Aturan Emas Perancangan *Interface*

Menurut Shneiderman (1998,p74-75), delapan aturan emas dalam perancangan *interface* yang interaktif, yaitu:

1. Berusaha keras untuk konsisten

Beberapa bentuk konsistensi yang ada adalah konsistensi dalam penggunaan bentuk dan ukuran *Font*, pemberian warna tulisan dan latar belakang, pembuatan layout,

penggunaan terminologi. Konsistensi memberikan kemudahan bagi pengguna dalam menggunakan dan menjalankan aplikasi, membantu mempelajari hal baru lebih cepat dan dapat lebih memfokuskan pada suatu tugas karena pengguna tidak perlu membuang waktu untuk mengingat perbedaan-perbedaan dalam interaksi.

2. *Memungkinkan pengguna untuk sering menggunakan jalan pintas (Shortcut)*

Jika frekuensi penggunaan meningkat, maka perlu dilakukan pengurangan jumlah interaksi dengan memanfaatkan fasilitas *shortcut*. *Shortcut* seperti tombol-tombol khusus, *Hidden Commands* dan fasilitas makro disukai *Frequent Knowledgeable User* karena dapat memberikan waktu respon yang semakin singkat dan waktu tampilan makin cepat.

3. *Memberikan umpan balik yang informatif*

Sistem harus menyediakan umpan balik untuk setiap aksi pemakai. Isyarat-isyarat seperti suara dan tampilan visual harus ditampilkan untuk setiap interaksi pemakai untuk memberi tahu bahwa aplikasi memberikan respon masukan dari pemakai.

4. *Merancang dialog untuk menghasilkan keadaan akhir*

Urutan aksi harus tersusun dalam bagian awal, tengah, dan akhir suatu grup. Umpan balik penyelesaian aksi yang informatif, memberikan kepada pengguna, perasaan lega, sinyal untuk mendapatkan kemungkinan rencana dan pilihan dari pikiran pengguna, serta indikasi bahwa cara tersebut jelas mempersiapkan aksi berikutnya.

5. *Menyediakan pencegahan kesalahan dan penanganan kesalahan yang sederhana*

Penggunaan desain antarmuka yang terbaik pun, pemakai tetap dapat membuat kesalahan. Kesalahan ini dapat secara fisik (secara tidak sengaja merujuk ke perintah dan data yang salah) dan secara mental (membuat keputusan yang salah mengenai perintah dan data yang dipilih). Maka sistem didesain sedemikian rupa agar

pengguna tidak membuat kesalahan serius. Jika terjadi kesalahan, sistem harus bisa mendeteksi dengan menawarkan mekanisme penanganan yang sederhana dan mudah dimengerti.

6. *Mengijinkan pembalikan aksi dengan mudah*

Sedapat mungkin aksi harus dapat diulang. Hal ini dapat mengurangi, menghilangkan kecemasan karena pengguna tahu bahwa kesalahan dapat dihilangkan. Dengan begitu mendorong kita untuk menjelajahi pilihan-pilihan yang tidak dikenal.

7. *Pengontrolan terletak pada pengguna sendiri*

Pengguna harus merasa bahwa ia berusaha atas sistem dan sistem menuruti perintahnya. Tindakan sistem yang mengejutkan, data yang rumit, kesalahan dalam memperoleh informasi penting dan ketidakmampuan untuk melakukan tindakan yang diinginkan akan membuat kecemasan dan ketidakpuasan.

8. *Mengurangi beban ingatan jangka pendek*

Manusia mempunyai keterbatasan dalam menghafal. Maka tampilan sebaiknya dibuat sederhana, tampilan informasi dikelompokkan, frekuensi gerakan window dikurangi, serta diberikan waktu latihan yang cukup untuk kode, *Mnemonics* dan urutan aksi. Untuk lebih memudahkan, sebaiknya dibuat akses *online* untuk mencari istilah, singkatan maupun urutan aksi yang sering dipakai namun sulit dihafal.

2.5 Basis Data

Menurut Connolly dan Begg (2002, p14) *basis data* adalah kumpulan data (dan gambaran data tersebut) yang terhubung secara logika dan dapat dipakai bersama, yang dirancang untuk memenuhi kebutuhan informasi dari sebuah perusahaan.

Pada sub bab ini, akan dibahas mengenai pengertian dari sistem basis data, beserta komponen-komponennya, relasi dari sistem basis data dan contoh dari bahasa pemrograman database (pada kasus ini penulis menggunakan bahasa pemrograman *mySQL*).

2.5.1 Sistem Basis Data

Sistem basis data merupakan sistem penyimpanan informasi yang terorganisasikan secara komputerisasi sehingga memudahkan pemakai dalam pengolahan dan penggunaannya. Tujuan hal tersebut adalah untuk melakukan perawatan informasi dan menyajikannya kapan saja dibutuhkan oleh pengguna.

Sistem basis data dapat dibagi menjadi empat komponen, yaitu:

1. Data yang bergantung pada konfigurasi sistem. Dalam sistem besar, data harus terintegrasi dan dapat dipakai bersama.
2. Perangkat keras yang diperlukan manajemen basis data dalam mengolah atau menyimpan database.
3. Piranti lunak yang menghubungkan antara fisik basis data dengan pengguna yaitu sistem manajemen basis data (*DBMS*).
4. Pengguna yang menggunakan data secara optimal seperti pemrograman piranti lunak untuk mengakses *database end user* yang menggunakan piranti lunak dalam mengakses basis data dan *administrator* yang bertanggung jawab terhadap keseluruhan basis data.

2.5.2 Database Relational

Database (basis data) adalah sekumpulan data yang terdiri dari satu atau lebih tabel yang saling berhubungan. *Database Relational* merupakan *database* yang dalam gambaran penggunaannya merupakan kumpulan dari tabel-tabel, dimana tabel tersebut terdiri dari baris dan kolom atau dengan kata lain terdiri dari kumpulan *record* dan *field*. Tabel tersebut kemudian dihubungkan dengan satu *field* yang disebut sebagai *key*. Pada *database relational* terdapat dua jenis *field* yang dapat menggambarkan hubungan, yaitu *primary key* dan *foreign key*.

Primary key adalah suatu *field* yang menghubungkan satu tabel dengan tabel yang lainnya. *Foreign key* merupakan sebuah *field* yang digunakan sebagai *field* tujuan yang dihubungkan dengan *field* dari tabel pemanggil.

Jenis hubungan *Database Relational* dapat dibagi menjadi tiga, yaitu:

1. Hubungan *one to one*, dimana satu isi *record* pada satu tabel hanya dapat berhubungan dengan satu *record* pada tabel lainnya.
2. Hubungan *one to many* atau *many to one*, dimana satu isi *record* pada satu tabel dapat berhubungan dengan beberapa *record* pada tabel lainnya atau sebaliknya.
3. Hubungan *many to many*, dimana satu isi *record* pada tabel A dapat berhubungan dengan beberapa *record* pada tabel B. Dan sebaliknya satu isi *record* pada tabel B dapat berhubungan dengan beberapa *record* pada tabel A.

2.5.3 Database Management System (DBMS)

Menurut Connolly dan Begg (2002, p16) sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, dan memelihara basis data dan juga menyediakan akses yang terkontrol ke basis data tersebut.

Fasilitas-fasilitas yang disediakan DBMS:

1. *Data Definition Language* (DDL)

DDL memungkinkan pengguna untuk menspesifikasikan tipe dan struktur data, dan batasan pada data yang tersimpan pada basis data.

2. *Data Manipulation Language* (DML)

DML memungkinkan pengguna untuk memasukan, memperbaharui, menghapus dan mengambil data dari basis data.

2.5.4 MyStructured Query Language (MySQL)

MySQL (Weiling dan Thomson, 2001, p2) adalah *Relational Database Management System* (RDBMS) yang sangat cepat dan kuat.

MySQL (Maslakowski dan Butcher, 2000, p10) adalah DBMS yang bersifat relasional, *open source*, berlevel *enterprise*, *multithread*.

MySQL merupakan bahasa yang memiliki kemampuan cukup baik untuk menunjang kerja *user*, baik yang telah berpengalaman dengan *database* maupun untuk pemula. *SQL* digunakan untuk mencari informasi (*query*), memanipulasi data(*DML*), mendefinisikan data (*DDL*), dan bahasa pengendali dokumentasi.

Setiap pengguna basis data memerlukan bahasa pemrograman yang dapat dipakai sesuai dengan fungsi dan tugasnya. Dalam basis data secara umum dikenal dua bahasa, yaitu:

1. *Data Definition Language* (*DDL*): bahasa yang dipakai untuk menjelaskan objek dari basis data. *DDL* dipakai untuk mendefinisikan kerangka basis data (berorientasi pada tipe pada objek basis data).

2. *Data Manipulation Language (DML)*: bahasa yang dipakai untuk memanipulasi objek data dari basis data. *DML* dipakai untuk operasi terhadap isi basis data.

2.6 PHP

PHP adalah bahasa *server-side scripting* yang digunakan untuk aplikasi *web* yang dinamis dan interaktif (Anonimus2, 2006). Sebuah halaman PHP adalah sebuah halaman *HTML* yang memiliki *server-side scripts* yang ditempatkan dalam *server* dan diproses oleh *webserver* sebelum dikirim ke *browser* pemakai.

Server-side scripts dijalankan ketika *browser* melakukan permintaan *file.php* dari *server*. PHP dipanggil oleh *webserver*, dimana proses *script* perintah yang ada di suatu halaman dieksekusi mulai dari awal sampai akhir di dalam mesin PHP. Setelah *script* PHP tersebut diolah, hasilnya akan ditampilkan kepada *client* melalui *web browser* berupa tampilan HTML. Penulisan *script* PHP diawali tanda `<?` Disisipi sintaks `”php”` dan diakhiri tanda `?>`.

Kelebihan-kelebihan dari PHP:

1. Kecepatan akses yang tinggi
2. Dapat bekerja dalam *webserver* yang berbeda dan sistem operasi yang berbeda
3. PHP adalah *freeware* dan *open source*
4. Merupakan bahasa pemrograman yang *embedded*
5. Dapat berjalan pada berbagai *platform* : Apache, IIS, Microsoft Personal WebServer

Alasan mengapa menggunakan PHP dan berbasis *web* adalah:

1. Bahasa pemrograman PHP terbukti sangat handal dalam membangun sebuah program berbasis *web*.

2. Waktu yang digunakan untuk memproses data dan menjalankan perintah-perintah *query* sangat cepat.
3. Dengan berjalan dalam sebuah *webserver*, maka secara otomatis program ini bersifat *multiuser*.
4. *Database* MySQL menyimpan data di dalam direktori khusus yang terpisah dari file program PHP sehingga keamanan data lebih terjamin.
5. *Web server* dan *database server* terpisah sehingga menyulitkan pihak luar untuk mengakses data yang terdapat di dalam *database*.
6. Bahasa program PHP dan *database* MySQL lebih fleksibel karena dapat diakses oleh sistem operasi Windows maupun Linux.
7. Program dapat diakses dari komputer manapun tanpa harus menginstall program *client*. Program bantuan untuk mengakses system ini hanyalah sebuah *browser*.