

BAB III

LANDASAN TEORI

3.1. Pembelian

Menurut Sofjan Assauri (2008:223) pembelian merupakan salah satu fungsi yang penting dalam berhasilnya operasi suatu perusahaan. Fungsi ini dibebani tanggung jawab untuk mendapatkan kuantitas dan kualitas bahan-bahan yang tersedia pada waktu dibutuhkan dengan harga yang sesuai dengan harga yang berlaku. Pengawasan perlu dilakukan terhadap pelaksanaan fungsi ini, karena pembelian menyangkut investasi dana dalam persediaan dan kelancaran arus bahan ke dalam pabrik.

Sedangkan menurut Mulyadi (2007:711) aktivitas dalam proses pembelian barang adalah:

1. Permintaan pembelian
2. Pemilihan pemasok
3. Penempatan order pembelian
4. Penerimaan barang, dan
5. Pencatatan transaksi pembelian

Permintaan pembelian adalah contoh suatu aktivitas yang merupakan satuan pekerjaan yang ditujukan untuk memicu bagian pembelian melakukan pengadaan barang sesuai dengan spesifikasi dan jadwal sebagaimana yang dibutuhkan oleh pemakai barang. Penerimaan barang adalah contoh aktivitas tentang penerimaan kiriman dari pemasok sebagai akibat adanya order pembelian yang dibuat oleh bagian pembelian.

Jenis-jenis pembelian menurut sifatnya dibagi dalam tiga macam pembelian, yaitu:

1. *Hand-To-Mouth Buying* (Pembelian yang Teratur)

Yaitu pembelian yang didasarkan atas besarnya kebutuhan sekarang. Maksudnya adalah untuk mencegah kerugian/keburukan yang diakibatkan oleh adanya persediaan bahan yang berlebih di gudang dan penggunaan modal dapat digunakan dengan sebaik-baiknya.

2. *Speculative Purchasing* (Pembelian Spekulatif)

Yaitu pembelian yang tidak didasarkan karena perlunya bahan itu dipergunakan dalam proses produksi sekarang, tetapi didasarkan karena suatu motif untuk mendapatkan keuntungan akan naiknya harga bahan pada waktu yang akan datang.

3. *Forward Buying* (Pembelian Sebelumnya)

Yaitu pembelian untuk memenuhi tersedianya bahan mentah secara continue agar perusahaan tidak sampai terganggu aktivitasnya karena tidak tersedianya bahan baku pada waktunya.

Adapun fungsi yang terkait pembelian menurut Mulyadi (2001:299), adalah sebagai berikut :

1. Fungsi Gudang

Fungsi gudang dalam sistem informasi pembelian bertanggung jawab untuk mengajukan permintaan pembelian sesuai dengan posisi persediaan yang ada di gudang dan untuk menyimpan barang yang telah diterima oleh fungsi penerimaan.

2. Fungsi Pembelian

Bertanggung jawab untuk memperoleh informasi mengenai harga barang, menentukan pemasok yang dipilih dalam pengadaan barang dan mengeluarkan order pembelian kepada pemasok yang dipilih.

3. Fungsi Penerimaan

Bertanggung jawab untuk melakukan pemeriksaan terhadap jenis, mutu dan kuantitas barang yang diterima dari pemasok guna menentukan dapat atau tidaknya barang tersebut diterima oleh perusahaan.

4. Fungsi Akuntansi

Fungsi akuntansi yang terkait dalam transaksi pembelian adalah fungsi pencatatan utang dan fungsi pencatatan persediaan.

3.2. Program Aplikasi

Menurut penjelasan dari Jogiyanto (2005:112), Program merupakan ekspresi, pernyataan kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh komputer. Sedangkan Aplikasi adalah suatu penerapan, menyimpan sesuatu hal, data, permasalahan pekerjaan kedalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai - nilai dasar dari hal, data, permasalahan atau pekerjaan.

Maka pernyataan Jogiyanto (2005:113) tentang Program Aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer agar dapat melakukan pekerjaan sesuai dengan keinginan dari permasalahan pengguna.

3.3. Analisis Sistem

Menurut Hartono (2005:129) analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem.

Dalam tahap analisis sistem menguraikan suatu informasi yang utuh ke dalam bagian-bagian yang bermaksud untuk mengidentifikasi dan melakukan evaluasi permasalahan-permasalahan yang ada.

Didalam tahap analisis sistem terdapat terdapat beberapa langkah yang harus dilakukan antara lain sebagai berikut :

1. Mengidentifikasi Masalah
2. Memahami kerja sistem yang ada
3. Menganalisis sistem
4. Membuat laporan

3.4. Perancangan

Menurut Susanto (2004:331) dalam buku Sistem informasi Manajemen Konsep dan Pengembangannya mengatakan “perancangan adalah spesifikasi

umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis”

Menurut John Bruch dan Gray Grudnitski yang telah diterjemahkan oleh Hartono (2005:196) dalam buku Analisis dan Desain Sistem Informasi mengatakan “desain sistem dapat didefinisikan sebagai pengembangan, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah dari satu kesatuan yang utuh dan berfungsi”.

3.5 Sistem Flow

Sistem flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem di mana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat sistem flow sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang ada (Hartono, 1998:10)

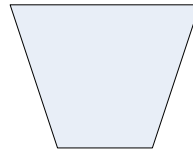
Terdapat berbagai macam bentuk symbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah terminator, manual operation, document, process, database, manual input, decision, off-line storage, on-page reference, dan off-page reference.

Terminator merupakan bentuk simbol yang di gunakan sebagai tanda di mulainya jalan proses sistem ataupun tanda akhir dari sebuah pengerjaan suatu sistem. Simbol dari *terminator* dapat dilihat pada gambar 3.1.



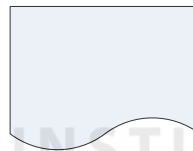
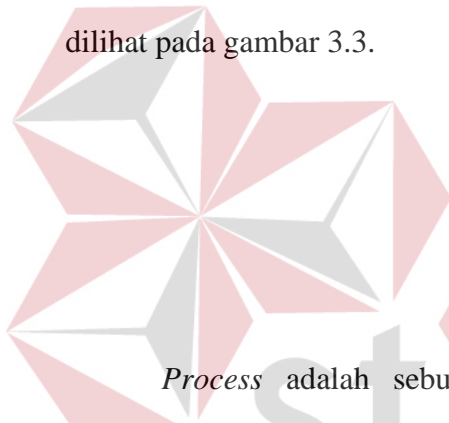
Gambar 3.1 *Terminator*

Manual operation digunakan untuk menggambarkan sebuah proses kerja yang dilakukan tanpa menggunakan computer sebagai medianya (menggunakan proses manual). Simbol dari *manual operation* dapat dilihat pada gambar 3.2.



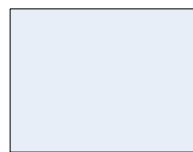
Gambar 3.2 *Manual Operation*

Document merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik. Simbol dari *document* dapat dilihat pada gambar 3.3.



Gambar 3.3 *Document*

Process adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. Simbol dari *process* dapat dilihat pada gambar 3.4.



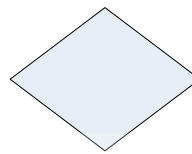
Gambar 3.4 *Process*

Database digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* dapat di lihat pada gambar 3.5.



Gambar 3.5 Database

Decision merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar dan salah. Simbol dari *decision* dapat dilihat pada gambar 3.6.



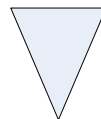
Gambar 3.6 Decision

Manual input digunakan untuk melakukan proses input ke dalam database melalui keyboard. Simbol dari *manual input* dapat dilihat pada gambar 3.7.



Gambar 3.7 Manual Input

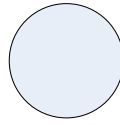
Off-line storage merupakan bentuk media penyimpanan yang berbeda dengan database, dimana media penyimpanan ini menyimpan dokumen secara manual atau lebih dikenal dengan nama arsip. Simbol dari *off-line storage* dapat dilihat pada gambar 3.8.



Gambar 3.8 Off – line Storage

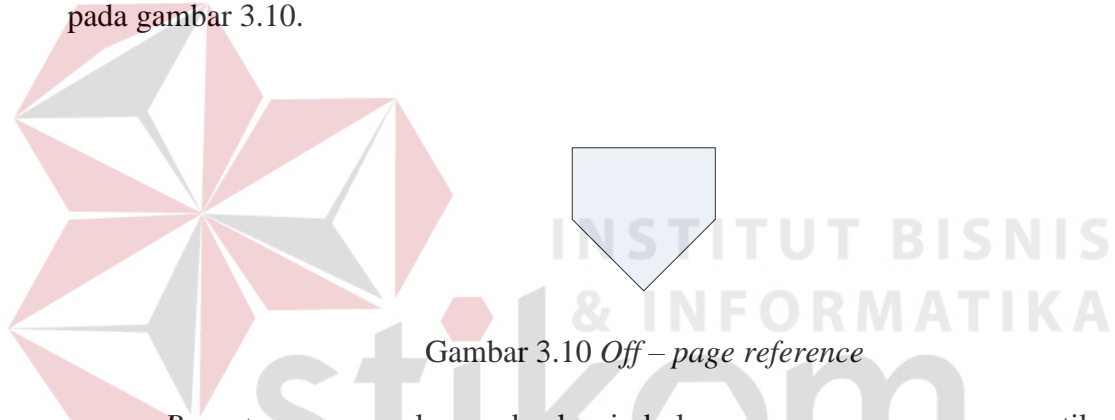
On-page reference digunakan sebagai simbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh

dalam permasalahan letaknya. Simbol dari *on-pagereference* dapat dilihat pada gambar 3.9.



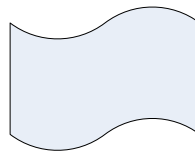
Gambar 3.9 *On – page reference*

Off-page reference memiliki sifat yang sedikit berbeda dengan on-page reference, karna simbol ini hanya digunakan apabila arus data yang ada dilanjutkan ke halaman yang berbeda. Simbol dari *off-page reference* dapat dilihat pada gambar 3.10.



Gambar 3.10 *Off – page reference*

Papertape merupakan sebuah simbol yang umumnya menggantikan bentuk penggambaran jenis pembayaran yang digunakan (misal : uang) dalam transaksi yang ada pada sistem yang di rancang. Simbol dari *paper tape* dapat dilihat pada gambar 3.11.



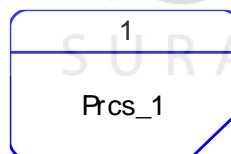
Gambar 3.11 *Paper Tape*

3.6 Data Flow Diagram (DFD)

Menurut (Kristanto, 2004:12), *Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana 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 (*structure analysis and design*). Penggunaan notasi dalam DFD sangat membantu untuk memahami suatu system pada semua tingkat kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

Didalam DFD, 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 dari *process* dapat dilihat pada gambar 3.12.



Gambar 3.12 *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 *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

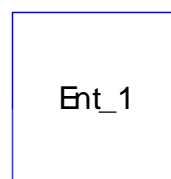
yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

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 di dalamnya. Pembuatannya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem, menentukan alur data yang diterima dan diberikan masing-masing proses dari pada 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), menggambar diagram level-0, menghindari perpotoingan 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 menggambar detail 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*, di mana 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. Simbol *external entity* dapat dilihat pada gambar 3.13.



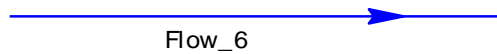
Gambar 3.13 *External Entity*

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 pelanggan, tabel detail penjualan, tabel detail pembelian, dan lain-lain. Simbol *data store* dapat dilihat pada gambar 3.14.



Gambar 3.14 *Data Store*

Data flow merupakan penghubung antar *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 didalam *data flow* mengandung sekumpulan data. Simbol *data flow* dapat dilihat pada gambar 3.15

Gambar 3.15 *Data Flow*

3.7 Entity Relationship Diagram (ERD)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD kita dapat menguji model dan mengabaikan proses apa yang harus dilakukan. ERD dapat dikategorikan menjadi tiga bagian, yaitu:

1. *One to one relationship*

Jenis hubungan antar tabel yang menggunakan bersama sebuah kolom primary key. Jenis hubungan ini tergolong jarang digunakan, kecuali untuk alasan keamanan atau kecepatan akses data. Misalnya satu departemen hanya mengerjakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.

2. *One to many relationship*

Jenis hubungan antar tabel dimana satu *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Jenis hubungan ini merupakan yang paling sering digunakan. Misalnya suatu pekerjaan hanya dikerjakan oleh satu departemen saja. Namun suatu departemen dapat mengerjakan beberapa macam pekerjaan sekaligus.

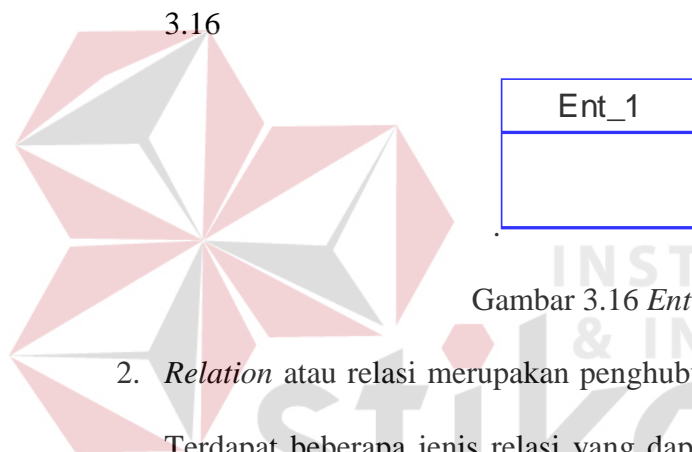
3. *Many to many relationship*

Jenis hubungan ini merupakan hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain.

Misalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.

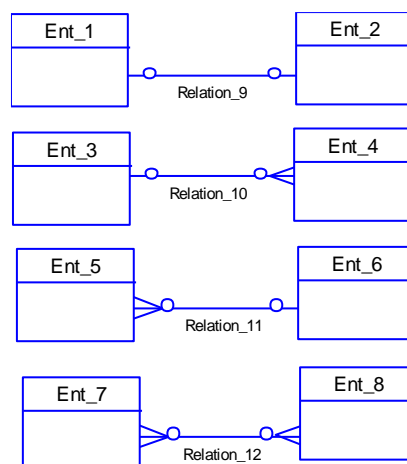
Entity Relationship Diagram, atau yang lebih dikenal dengan nama ERD, digunakan untuk mengimplementasikan, menentukan, dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari pemakai. Adapun elemen-elemen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

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



Gambar 3.16 *Entity*

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 *Relation Of Entity*

3.8 Program Penunjang

Adapun beberapa program penunjang yang digunakan untuk membantu dalam membuat aplikasi ini antara lain seperti berikut :

3.8.1. Visual basic .Net

Menurut Hidayatullah (2014:5) dalam buku Visual Basic .NET membuat aplikasi database dan program kreatif mengatakan visual basic .NET adalah visual basic yang direkayasa kembali untuk digunakan pada platform .NET sehingga aplikasi yang dibuat menggunakan visual basic .NET dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal .NET Framework.

3.8.2 Power Designer

Power designer merupakan suatu *tool* berupa *software* untuk mendesain sistem dan rancangan *Entity Relationship Diagram* (ERD) yang dikembangkan oleh *Sybase Inc.* Ada dua model data, yaitu : *Entity Relationship Diagram* (ERD) dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

- a. Model ERD atau *Conceptual Data Model* (CDM) : model yang di buat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (entity) serta hubungan (relationship) antara entitas-entitas itu.

- b. Model Relasional atau *Physical Data Model* (PDM) : model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik.

3.8.3 NET Framework

Microsoft .NET Framework (di baca Microsoft dot Net Framework) adalah sebuah komponen yang dapat ditambahkan ke sistem operasi *Microsoft Windows* atau telah terintegrasi ke dalam *Windows* (mulai dari *Windows server 2003* dan versi-versi *Windows* Terbaru). Kerangka kerja ini menyediakan sejumlah besar solusi-solusi program untuk memenuhi kebutuhan-kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk *framework* ini..*NET Framework* adalah kunci penawaran utama dari *Microsoft*, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk platform *Windows*.

Pada dasarnya, *.NET framework* memiliki 2 komponen utama :*CLR* dan *.NET Framework Class Library*.

Program-program yang ditulis untuk *.NET Framework* dijalankan pada suatu lingkungan *software* yang mengatur prsyarat-persyaratan *runtime program*.*Runtime environment* ini, yang juga merupakan suatu bagian dari *.NET Framework*, dikenal sebagai *Common Language Runtime* (CLR). CLR menyediakan penampilan dari *application virtual machine*, sehingga para *programmer* tidak perlu mengetahui kemampuan CPU tertentu yang akan menjalankan program. CLR juga menyediakan layanan-layanan penting lainnya

seperti jaminan keamanan, pengaturan memori, *garbage collection* dan *exception handling*/penanganan kesalahan pada saat *runtime*. *Class Library* dan CLR ini merupakan komponen inti dari *.NET Framework* .kerangka kerja itupun dibuat sedemikian rupa agar para *programmer* dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga computer dari beberapa ancaman keamanan.

CLR adalah turunan dari CLI (*Common Language Infrastructure*) yang saat ini merupakan standar ECMA.Untuk keterangan lebih lanjut, silahkan mengunjungi situs ECMA atau kunjungi sumber pranala dibawah artikel ini.

Solusi-solusi program pembentuk *Class Library* dari *.NET Framework* melindungi area yang luas dari kebutuhan program pada bidang *user interface*, pengaksesan data, koneksi basis data, kriptografi, pembuatan aplikasi berbasis web, algoritma numerik, dan komunikasi jaringan. Fungsi-fungsi yang ada dalam *class library* dapat digabungkan oleh *programmer* dengan kodenya sendiri untuk membuat suatu program aplikasi baru

3.8.4 Crystal Report

Merupakan *software* yang digunakan untuk pembuatan laporan. Dengan cara mengoneksi nama tabel yang akan dibuatkan laporannya. Setelah tampilan data ada maka klik dan drag semua *field* yang ada sesuai dengan tampilan yang diinginkan.Biasanya *crystal report* adalah komponen dari *VB.NET*

3.8.5 SQL server

Menurut Santoso (2006:5) mengatakan “sql server adalah hasil kerja sama antara Microsoft dengan Sybase untuk memproduksi sebuah software penyimpanan data (database) yang bekerja pada sistem Operasi OS/2”. Menurut Komputer (2010:2) dalam buku Short Course Series :SQL Server 2008 Express mengutip “SQL Server adalah DBMS (Database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle”

3.8.6 Microsoft Office Visio

Menurut Cato Chandra dan Ir. Teddy Marcus Zakaria (2007 :197) Microsoft Visio 2007 merupakan suatu program aplikasi komputer yang didesain khusus untuk membantu dalam membuat diagram seperti Flowchart, Grantt Chart, Data Flow, Gambar Jaringan, Gambar Denah Bangunan, dan juga pembuatan Gambar Teknik, Gambar Elektronik, serta desain lainnya. Microsoft Visio 2007 memiliki keunggulan dibanding program word atau excell dalam pembuatan diagram. 28 Microsoft Visio 2007 memiliki banyak template yang di dalam terdapat shape / konektor. Ada lima langkah utama untuk membuat sebuah diagram di Microsoft Visio 2007 yaitu :

- 1 Tentukan template yang sesuai.
- 2 Buat shape pada halaman kerja (page).
- 3 Buat konektor untuk menyambung anatar shape.
- 4 Buat teks (keterangan shape atau konektor).
- 5 Simpan dokumen visio tersebut.