BAB III

LANDASAN TEORI

3.1 Konsep Dasar Sistem Informasi

Menurut Jogianto (1999), terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu sistem yang menekankan pada prosedurnya dan menekankan pada proseduir mendefinisikan suatu sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu.

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*inputan*), keluaran (*output*), pengolah (*process*), dan sasaran (*objective*) atau tujuan (*goal*).

Komponen sistem merupakan baagian-bagian dari sistem yang saling berhubungan dan menjadi satu kesatuan. Komponen-komponen sistem atau sub-sub sistem ini memiliki karakteristik tersendiri dan menjalankan suatu fungsi tersendiri. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan *supra system* (Jogianto, 1999). Misal sebuah perusahaan dapat disebut sebagai sistem dan industri yang merupakan sistem yang lebih besar dapat disebut sebagai *supra system.*

Batas sistem *(boundary)* merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope)* dari sistem tersebut (Jogianto, 1999:4).

Lingkungan luar *(environment)* dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energy dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang merugikan harus

ditahan dan dikendalikan, agar tidak mengganggu kehidupan dari sistem itu sendiri (Jogianto, 1999).

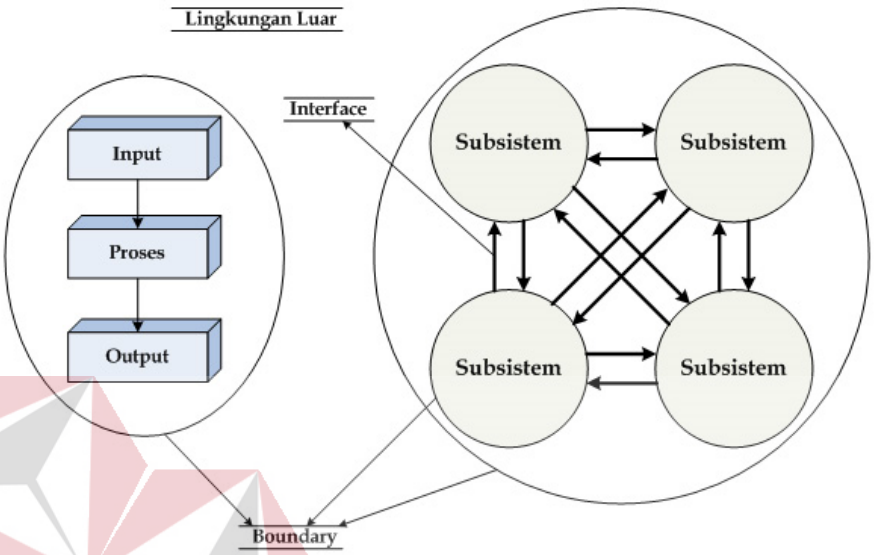
Penghubung *(interface)* merupakan media penghubung antara satu sub-sistem dengan sub-sistem yang lainnya. Melalui penghubung ini memungkinkan sumber daya-sumber daya mengalir dari suatu sub-sistem ke sub-sistem yang lainnya. Keluaran *(output)* dari suatu sub-sistem akan menjadi masukan *(input)* untuk sub-sistem yang lainya melalui penghubung *(interface).* Dengan penghubung *(interface),* suatu sub-sistem dapat berintegrasi dengan sub sistem lainya untuk membentuk suatu kesatuan (Jogianto, 1999:4-5).

Masukan *(input)* adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa sinyal atau berupa masukan perawatan. Masukan sinyal adalah energi yang dimasukkan yang nantinya akan diolah dan menghasilkan sesuatu. Sedangkan masukan perawatan adalah energi yang digunakan untuk melakukan suatu proses atau dengan kata lain energi yang menjamin suatu proses dapat berjalan. Keluaran sistem dapat dibedakan menjadi dua yaitu keluaran yang berguna dan sisa pembuangan. Keluaran dapat dijadikan sebagai masukan dari sub-sistem yang lainnya (Jogianto, 1999:5).

Pengolah sistem *(process)* adalah bagian dari setiap sistem dan sub-sistem yang akan mengolah masukan sehingga menjadi keluaran *(output),* baik yang berguna maupun menjadi sisa (Jogianto, 1999:5).

Suatu sistem pasti mempunyai tujuan (*goal*) ataupun sasaran (*objective*) yang ingin dicapai. Jika suatu sistem tidak memiliki sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran sistem sangan menentukan apa yang diperlukan serta keluaran apa yang dihasilkan. Suatu sistem dikatakan berhasil jika mengenai sasaran yang ingin dicapai (Jogianto, 1999:5).

Karakteristik dari suatu sistem dapat digambarkan dalam bagan sebagai berikut :



Gambar 3.1 Karakteristik suatu sistem

Informasi dapat diibaratkan sebagai darah dalam suatu mahluk hidup. Informasi member suatu semangat, motivasi, dan gairah dalam suatu organisasi. Tanpa adanya informasi, organisasi tersebut akan lesu, kerdil, dan akhirnya akan berhenti. Menurut Jogianto, informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi itu sendiri adalah data, yang merupakan jamak dari bentuk tunggal *datum.* Data adalah kenyataan yang menggambarkan suatu keadaan nyata. Data juga merupakan bentuk yang masih mentah dan belum dapat bercerita banyak, sehingga membutuhkan pengolahan yang lebih lanjut. Kualitas dari sistem informasi bergantung pada dua hal, yaitu:

1. Informasi harus akurat, dimana informasi itu harus bebas dari kesalahan.
2. Informasi tersebut harus relevan, supaya informasi tersebut dapat memberikan masukan bagi penerimanya.

Secara keseluruhan Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan pihak luar dengan laporan-laporan yang diperlukan.

Istilah sistem informasi juga sering di kacaukan dengan sistem informasi manajemen (SIM). Kedua hal tersebut sebenarnya tidak sama. Sistem informasi manajemen (SIM) merupakan kumpulan dari sistem-sistem yang menyediakan informasi untuk mendukung manajemen .

3.2 Analisis dan Perancangan sistem

Analisis sistem merupakan tahap yang paling penting dari suatu pengembangan sistem karena merupakan tahap awal untuk melakukan evaluasi permasalahan yang terjadi serta kendala-kendala yang dihadapi dari sebuah sistem yang telah berjalan.

Analisis sistem itu sendiri dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluai permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Jogianto, 1999).

Analisis yang efektif akan memudahkan pekerjaan penyusunan rencana yang baik di tahap berikutnya. Sebaliknya, kesalahan yang terjadi pada tahap analisis ini akan menyebabkan kesulitan yang lebih besar, bahkan menyebabkan gagalnya penyusunan sebuah sistem .

Untuk itu, diperlukan ketelitian dalam mengerjakan, sehingga tidak dapat kesalahan dalam tahap selanjutnya, yaitu tahap perancangan sistem. Langkah-langkah yang diperlukan di dalam menganalisa sistem adalah:

1. Tahap perencanaan sistem
2. Tahap analisis sistem
3. Tahap perancangan sistem

4. Tahap penerapan sistem

5. Membuat laporan dari hasil analisa

Pada tahap perancangan, dilakukan identifikasi masalah serta diperlukan adanya analisa yang digunakan untuk menentukan faktor-faktor yang menjadi permasalahan dalam sistem yang telah ada atau digunakan.

Data-data yang baik yang berasal dari sumber-sumber internal seperti misalnya laporan-laporan, dokumen observasi, maupun sumber-sumber di luar lingkungan sistem seperti pemakai sistem, dikumpulkan sebagai bahan pertimbangan analisa. Jika semua permasalahan sudah di identifikasi, dilanjutkan dengan mempelajari dan memahami alur kerja dari sistem yang digunakan.

Kemudian diteruskan dengan menganalisa dan membandingkan sistem yang terbentuk dengan sistem yang sebelumnya di gunakan. Dengan adanya perubahan tersebut, maka langkah selanjutnya adalah membuat laporan-laporan hasil analisasebelumnya dan sistem yang akan diterapkan. Perancangan sistem adalah proses menyusun atau mengembangkan sistem informasi yang baru. Dalam tahap ini, harus dipastikan bahwa semua persyaratan untuk menghasilkan informasi dapat terpenuhi.

Hasil sistem yang dirancang harus sesuai dengan kebutuhan pemakai, karena rancangan tersebut meliputi perancangan mulai dari sistem yang umum hingga diperoleh sitem yang lebih spesifik. Dari hasil rancangan tersebut, dibentuk pula rancangan *database* disertai dengan struktur *file* antara sistem yang satu dengan sistem yang lainnya. Selain itu, dibentuk pula rancangan *input* dan *output* system, misalnya menentukan berbagai bentuk *input* data dan isi laporan.

Apabila di dalam perancangan sistem terdapat kesalahan maka kita perlu melihat kembali dari analisa sistem yang telah dibuat. Sehingga dapat di ambil kesimpulan bahwa analisa sistem mempunyai hubungan erat dengan perancangan sebuah sistem.

3.3 System Flow

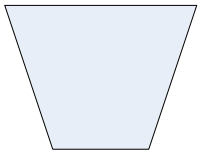
System Flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi-fingsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang aa (Jogianto, 1998:10).

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

Terminator merupakan bentuk symbol yangdigunakan sebagai tanda dimulainya jalan proses ataupun tanda akhir dari sebuah pengerjaan suatu sistem. Simbol dari *terminator* dapat dilihat pada Gambar.

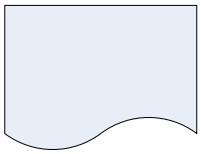
Gambar 3.2 *Terminator*

Manual operation digunakan untuk menggambarkan sebuah proses kerja yang digunakan tanpa menggunakan komputer sebagai medianya (menggunakan proses manual). Simbol dari *manual operation* dapat dilihat pada Gambar.



Gambar 3.3 *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.



Gambar 3.4 *Document*

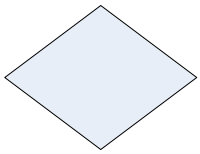
Process adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. Simbol dari *process* dapat dilihat pada Gambar.

Gambar 3.5 *Process*

Database digunakan sebagai media penyimpanan data yang besifat terkomputerisasi. Simbol dari *database* dapat dilihat pada Gambar.

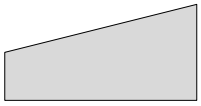
Gambar 3.6 *Database*

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



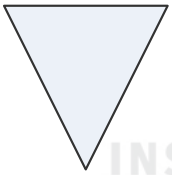
Gambar 3.7 *Decision*

Manual input digunakan untuk melakukan proses *input* kedalam *database* melalui *keyboard.* Simbol dari *manual input* dapat dilihat pada Gambar.



Gambar 3.8 *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 storege* dapat dilihat pada Gambar.

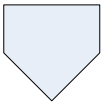


Gambar 3.9 *Off-Line Storage*

On-page reference digunakan sebagai symbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh dalam permasalahan letaknya. Simbol dari *on-page reference* dapat dilihat pada Gambar.

Gambar 3.10 *On-Page Reference*

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



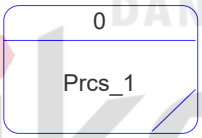
Gambar 3.11 *Off-Page Reference*

3.4 *Data Flow Diagram* (DFD)

Menurut (Kristanto, 2004), *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 meerupakan suatu metode pengembangan sistem yang tersetruktur *(structure analisys dan design)*. Penggunaan notasi dalam data flow diagram sangat membantu untuk memahami suatu sistem pada semua tingkat kompleksitas. Pada tahap analisi, penggunaan notasi ini dapat membantu dalam berkimunikasi 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.

a. *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:

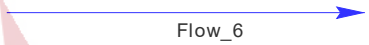
Gambar 3.13 *Simbol External Entity*

b. *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 betuk 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*

c. *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*

3.5 *Entity Relationdhip Diagram* (ERD)

Proses *reverse engineering* terhadap suatu basis data menjadi suatu kebutuhan bagi perancang basis data untuk mengetahui struktur dari sebuah basis data. Struktur tersebut biasanya dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD).

ERD dibagi menjadi 2 macam yaitu: *Conceptual Data Model* (CDM), dan *Physical Data Model* (PDM). Simbol-simbol yang sering digunakan adalah:

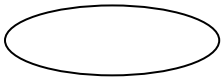
1. *Entity*

Entity merupakan sesuatu yang mudah diidentifikasikan. Sebuah *entity* bisa berupa obyek, tempat, orang, konsep, atau aktivitas. *Entity* dinyatakan dalam simbol persegi panjang. Simbol *entity* pada Gambar 3.5.

Gambar 3.16 *Entity*.

2. Atribut

Atribut merupakan penjelasan-penjelasan dari *entity* yang membedakan *entity* satu dengan yang lain. Sebuah atribut juga merupakan sifat-sifat dari sebuah *entity*. Atribut dinyatakan dalam simbol ellips. Simbol atribut pada Gambar 3.6.



Gambar 3.17 *Atribut*.

3. Relationship

Relationship adalah penghubung antara suatu *entity* dengan *entity* yang lain dan merupakan bagian yang sangat penting di dalam mendesain *database*. Ada empat tipe *relationship* yang dikenal yaitu :

a. One-to-One Relationship

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

b. One-to-Many Relationship

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

c. Many-to-Many Relationship

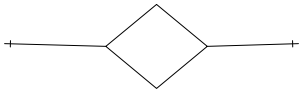
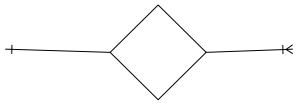
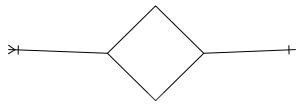
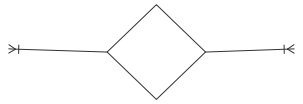
Jenis hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Miasalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.

d. Many-to-One Relationship

Jenis hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan satu *record* pada tabel lain. Miasalnya satu departemen mampu mengerjakan banyak pekerjaan, nemun satu pekerjaan hanya dikerjakan oleh satu departemen saja.

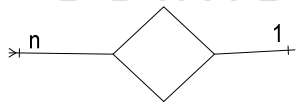
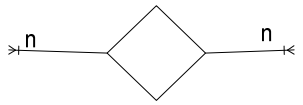
Menurut Sutanta (2004), relasi antar entitas dapat digambarkan melalui salah satu dari pilihan di bawah ini:

1. Pilihan 1

Jenis relasi	Simbol yang digunakan
1-ke-1 :	
1-ke-n :	
n-ke-1 :	
n-ke-n :	

Gambar 3.18 Simbol relasi antar entitas (pilihan 1).

2. Pilihan 2

Jenis relasi	Simbol yang digunakan
1-ke-1 :	
1-ke-n :	
n-ke-1 :	
n-ke-n :	

Gambar 3.19 Simbol relasi antar entitas (pilihan 2).

ERD dapat digambarkan menggunakan salah satu dari pilihan di atas, namun penggunaannya harus konsisten. Jika menggunakan simbol pilihan 1, maka untuk seluruh bagian ERD harus menggunakan simbol kelompok pilihan 1.

3. Kunci relasi

Kunci relasi atau *key* adalah suatu properti yang menentukan apakah suatu kolom pada table sangat penting atau tidak. Berdasarkan macamnya, kunci relasi terdiri dari:

a. Kunci kandidat

Yaitu satu atau atau gabungan minimal atribut yang bersifat unik yang dapat digunakan untuk mengidentifikasi setiap *record* dalam relasi.

b. Kunci primer

Yaitu bagian atau salah satu dari kunci kandidat yang digunakan sebagai kunci utama untuk membedakan setiap *record* dalam relasi. Kunci primer biasa disebut sebagai *primary key*.

c. Kunci alternatif

Yaitu bagian dari kunci kandidat yang tidak digunakan sebagai kunci utama.

d. Kunci penghubung

Kunci penghubung atau *foreign key* yaitu satu atau gabungan sembarang atribut yang menjadi kunci utama dalam relasi lain yang mempunyai hubungan secara logik. Kunci penghubung dan kunci utama harus memiliki tipe dan ukuran data yang sama.

3.6 Program Penunjang

3.6.1 Visual Basic .NET

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi bergerak diatas sistem *.NET Framwork*, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini para pembuat program dapat membangun aplikasi *Windows Forms*. Alat ini bias diperoleh secara terpisah dari produk lainnya (seperti *Microsoft Visual C++, Visual C#, atau Visual J#*), atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*. Bahasa *Visual Basic .NET* sendiri menganut paradgma baha pemrograman berbasis objek yang dilihat sebagai evaluasi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan di atas *.NET Framwork*. Peluncuran mengandung kontraversi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.

3.6.2 *.NET Framework*

Microsoft .NET Framework (dibaca 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 yaitu *Common Language Runtime* (CLR) dan *.NET Framework Class Library*.

Program - program yang ditulis untuk *.NET Framework* dijalankan pada suatu lingkungan software yang mengatur persyaratan-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 itu pun dibuat sedemikian rupa agar para programmer dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga komputer dari beberapa ancaman keamanan.

3.6.3 SQL Server 2005

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri 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 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. Pada versi sebelumnya, MS SQL Server 2000 terserang oleh cacing komputer SQL Slammer yang mengakibatkan kelambatan akses Internet pada tanggal 25 Januari 2003.

3.6.4 *Crystal Report*

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