

BAB III

LANDASAN TEORI

3.1 Sistem

Menurut Jogiyanto, H.M (1989 : 23), menyebutkan bahwa sistem merupakan kumpulan dari elemen-elemen yang satu dengan yang lain berinteraksi dan bersama-sama beroperasi untuk mencapai tujuan tertentu. Sistem mempunyai peran yang sangat besar dalam menentukan berjalan tidaknya suatu lembaga atau perusahaan. Hal ini dikarenakan setiap perusahaan akan selalu berdasarkan pada suatu sistem dalam menjalankan aktifitas sehari-harinya.

Suatu sistem dapat dirumuskan sebagai suatu totalitas himpunan yang terdiri dari bagian-bagian yang mana antara satu dengan yang lainnya saling berinteraksi dan bersama-sama beroperasi guna mencapai suatu tujuan tertentu didalam suatu lingkungan. Bagian-bagian atau subsistem tersebut merupakan suatu kompleksitas tersendiri, tapi dalam kebersamaan mencapai suatu tujuan berlangsung secara harmonis dalam keteraturan yang pasti.

3.2 Sistem Informasi

Menurut Hartono (1999 : 11), sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolah transaksi atau informasi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Kegiatan dalam sistem informasi mencakup :

1. *Input*, menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
2. *Process*, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, suatu kegiatan untuk menghasilkan laporan dari proses diatas tersebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
5. *Control*, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

3.3 Sistem Penggajian

Gaji dan upah merupakan bagian dari kompensasi-kompensasi yang paling besar yang diberikan perusahaan sebagai balas jasa kepada karyawannya. Dan bagi karyawan ini merupakan nilai hak dari prestasi mereka, juga sebagai motivator dalam bekerja.

Sedangkan bagi perusahaan jasa, gaji dan upah merupakan komponen biaya yang mempunyai dampak besar dalam mempengaruhi laba, sehingga harus terus menerus diawasi pengelolaannya.

Niswonger (1999 : 446) mengemukakan bahwa :

“Istilah gaji (salary) biasanya digunakan untuk pembayaran atas jasa manajerial, administratif, dan jasa-jasa yang sama. Tarif gaji biasanya diekspresikan dalam periode bulanan. Istilah upah (wages) biasanya digunakan untuk pembayaran kepada karyawan lapangan (pekerja kasar) baik yang terdidik maupun tidak terdidik. Tarif upah biasanya diekspresikan secara mingguan atau perjam.”

Dari pendapat tersebut, dapat disimpulkan bahwa gaji merupakan balas jasa yang diberikan kepada karyawan yang mempunyai ikatan kerja kuat secara

berkala berdasarkan ketentuan yang berlaku di perusahaan dan sifatnya tetap. Sedangkan upah merupakan balas jasa yang di berikan kepada karyawan yang ikatan kerjanya kurang kuat berdasarkan waktu kerja setiap hari ataupun setiap minggu.

3.4 Sistem absensi

Absensi adalah suatu cara bagi perusahaan untuk melakukan *control* pada karyawan berkaitan dengan kehadiran. Dan juga, absensi berguna untuk melakukan penilaian terhadap karyawan mengenai kedisiplinan karyawan sehingga perusahaan dapat menetapkan kebijakan kepada karyawan.

3.5 Sistem pendaftaran

Pendaftaran merupakan proses awal untuk melakukan seleksi penerimaan karyawan baru. Pendaftaran dilakukan oleh calon karyawan kepada perusahaan dan diseleksi oleh bagian personalia. Proses pendaftaran juga bisa dijadikan acuan bagi perusahaan seberapa besar minat masyarakat untuk berkarir di-perusahaan.

3.6 Database

Database adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyiapkan atau merekam serta memelihara data operasional lengkap dengan sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan (Linda, 2004 : 1). *Database* dapat dinyatakan sebagai suatu sistem yang memiliki karakteristik seperti berikut :

1. Merupakan suatu kumpulan interaksi data yang disimpan bersama dan tanpa mengganggu satu sama lain atau membentuk duplikat data.
2. Kumpulan data di dalam *database* dapat digunakan oleh sebuah program secara optimal.
3. Penambahan data baru, modifikasi dan pengambilan kembali dari data dapat dilakukan dengan mudah dan terorganisasi.

Dalam arsitektur database terdapat tiga tingkatan yang saling mendukung.

Dibawah ini adalah penjelasannya yaitu :

1. *Internal level* yaitu tingkat yang basis datanya secara fisik ditulis atau disimpan di media *storage* dan level yang berkaitan.
2. *External level* disebut juga *indivisual user view*, yaitu tingkat yang basis datanya dapat berdasarkan kebutuhan masing-masing aplikasi di user atau level yang berkaitan dengan para pemakai.
3. *Conceptual level* disebut juga *community user view*, yaitu tingkat *user view* dari aplikasi yang berbeda digabungkan sehingga menggunakan basis data secara keseluruhan dengan menyembunyikan penyimpanan data secara fisik yang merupakan penghubung dari *internal level* dan *external level*.

Seluruh operasi yang dilakukan pada *database* didasarkan atas tabel-tabel dan hubungannya. Dalam model relasional dikenal antara lain *table*, *record*, *field*, *index*, *query*. Penjelasannya seperti dibawah ini :

1. *Table* atau *entity* dalam model relasional digunakan untuk mendukung antar muka komunikasi antara pemakai dengan profesional komputer.
2. *Record* atau baris atau dalam istilah model relasional yang formal disebut *tuple* adalah kumpulan data yang terdiri dari satu atau lebih.

3. *Field* atau kolom atau dalam istilah model relasional yang formal disebut *attribute* adalah sekumpulan data yang mempunyai atau menyimpan fakta yang sama atau sejenis untuk setiap baris pada *table*.
4. *Index* merupakan tipe dari suatu table tertentu yang bersis nilai-nilai *field* kunci atau *field*.
5. *Query* merupakan sekumpulan perintah *Structure Query Language* (SQL) yang dirancang untuk memanggil kelompok *record* tertentu dari satu tabel atau lebih untuk melakukan operasi pada tabel.

3.7 Sistem Aplikasi Komputer Berbasis Web

Aplikasi dalam bahasa awam sering disebut sebagai sebuah kumpulan program atau *script*. Aplikasi web yang dibangun dengan menggunakan *Struts framework* terdiri dari komponen-komponen individual yang digabungkan menjadi satu aplikasi. Aplikasi tersebut dapat diinstal dan dieksekusi oleh *web container*. Komponen-komponen tersebut dapat digabungkan karena mereka terletak dalam sebuah konteks *web* yang sama, yang menjadikan mereka bergantung satu dengan yang lainnya, baik secara langsung ataupun tidak langsung.

3.8 Microsoft Visual Basic.Net 2005

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 ke VB 2005, yaitu :

1. 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.
2. 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*.
3. *Windows Form Designer* memungkinkan *developer* memperoleh aplikasi *desktop* dalam waktu yang singkat.
4. 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*.
5. Visual Basic 2005 menghasilkan web. Menggunakan form web yang baru, anda dapat dengan mudah membangun *thin-client* aplikasi berbasis web.
6. Mendukung pembangunan aplikasi *client-server*, terdistribusi serta berupa berupa aplikasi yang berbasis *windows* serta web.
7. .NET Framework secara mendasar dibuat untuk dipasangkan dengan *windows* 2003 dengan keunggulan memonitor kelalaian dari aplikasi yang sedang berjalan, dan mmengisolasi setiap aplikasi yang sedang berjalan.
8. *Developer* dengan berbagai latar belakang dapat dengan segera menguasai .NET karena kemudahan dan kemiripan kode yang ditawarkannya.

9. *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.
10. Integrasi dengan sistem yang sudah ada sangat mudah, NET Framework com memungkinkan anda berinteraksi dengan sistem yang sudah ada menggunakan XML web service.
11. Mendukung lebih dari 20 bahasa pemrograman yang tidak terbayang sebelumnya. Memungkinkan pengembang memilih bahasa pemrograman yang tepat sesuai latar belakang pemrogramannya.

3.9 Microsoft 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 2000 service pack 4, windows XP profesional service pack 2, atau windows 2003 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*. (Wijayanti,2007 : 78)

3.10 Crystal Report

Crystal report adalah suatu form khusus berbentuk seperti lembaran format naskah yang ingin dicetak. Di dalam crystal report, kita dapat merancang laporan-laporan yang ingin kita tampilkan dari data-data yang terdapat di dalam *database*. Crystal report dapat berdiri sendiri, namun dapat juga menjadi satu dengan project visual basic yang dibuat atau dikembangkan. Bila berdiri sendiri, report tersebut-pun dapat dipanggil dari project visual basic dengan control Crystal Report Control sehingga report yang telah dibuat dapat digunakan oleh beberapa project sekaligus.

3.11 Microsoft SQL Server 2005 Express

Microsoft SQL Server merupakan produk RDBMS (Relational *Database* Management System) yang dibuat oleh Microsoft. Microsoft SQL Server juga mendukung SQL (Structured Query Language) sebagai bahasa untuk memproses baris perintah ke dalam basis data. SQL ini telah digunakan secara umum pada semua produk *database* server yang ada di pasaran saat ini. Microsoft SQL Server banyak digunakan pada dunia bisnis, pendidikan, dan juga pemerintahan sebagai solusi *database* atau media penyimpanan data. Berbagai macam skala bisnis, dari bentuk bisnis kecil sampai bisnis skala enterprise dapat menggunakan Microsoft SQL Server sebagai pusat basis datanya. Microsoft SQL Server merupakan sebuah *database* relational yang dirancang untuk mendukung aplikasi dengan arsitektur client-server, dimana *database* terdapat pada komputer pusat yang disebut dengan server, dan informasi digunakan bersama-sama oleh beberapa user

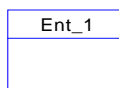
yang menjalankan aplikasi di dalam komputer lokalnya yang disebut dengan client. Arsitektur semacam ini memberikan integritas data yang cukup tinggi, karena semua user bekerja dengan informasi yang sama. Arsitektur client-server dapat mengurangi lalu lintas jaringan karena prosesnya hanya berjalan dengan permintaan data yang diperlukan oleh user.

Microsoft SQL Server 2005 Express dibagi kedalam beberapa komponen logis, seperti misalnya table, view, dan elemen-elemen lain yang dapat dilihat oleh user dengan menambahkan *add-on* dari aplikasi dengan nama *database management system*. Elemen-elemen ini secara fisik disimpan di dalam dua atau lebih file di dalam disk. Format file atau lokasi dimana elemen logic ini ditulis, tidak diketahui oleh user sistem. Apabila suatu *database* telah dibuat, user bisa memiliki akses yang telah diberikan kepadanya. Hal ini membuat Microsoft SQL Server 2005 Express dapat menyimpan beberapa *database* dan membatasi akses ke masing-masing *database* kepada user tertentu.

3.12 Entity Relationship Diagram

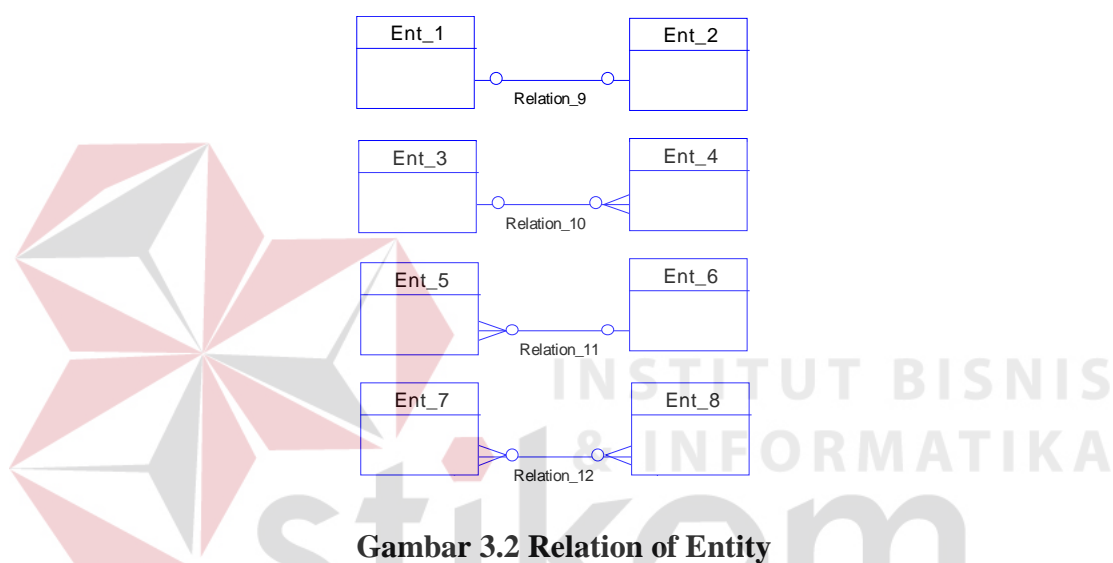
Entity Relationship Diagram, atau yang lebih dikenal dengan nama ERD, digunakan untuk mengimplementasikan, menentukan, dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari pemakai. Adapun elemen-elemen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

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



Gambar 3.1 Entity atau Entitas

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 berikut:



Gambar 3.2 Relation of Entity

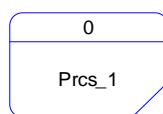
3.13 Data Flow Diagram

Menurut Andri Kristanto (2004), Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana 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 data flow diagram sangat membantu untuk memahami suatu sistem pada semua tingkat

kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi 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:



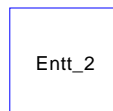
Gambar 3.3 Process

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), menggambar 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 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, dimana 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. External entity digambarkan dalam bentuk sebagai berikut:



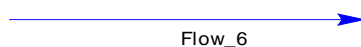
Gambar 3.4 External Entity

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 betuk data yang tersimpan pada data store tersebut, misalnya tabel pelanggan, tabel detail penjualan, tabel detail pembelian, dan lain-lain. Data store digambarkan dalam bentuk simbol sebagai berikut:



Gambar 3.5 Data Store

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 3.6 Data Flow

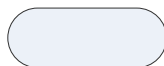
3.14 System Flow

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-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang ada (Jogiyanto, 1998).

Terdapat berbagai macam bentuk simbol 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 digunakan sebagai tanda dimulainya jalan proses sistem ataupun tanda akhir dari sebuah pengerjaan suatu sistem. Bentuk dari terminator adalah sebagai berikut:



Gambar 3.7 Terminator

Manual operation digunakan untuk menggambarkan sebuah proses kerja yang dilakukan tanpa menggunakan komputer sebagai mediana (menggunakan proses manual). Bentuk simbolnya adalah:



Gambar 3.8 Manual Operation

Document merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik. Bentuk dari document di gambarkan dalam simbol berikut:



Gambar 3.9 Document

Process adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. *Process* disimbolkan dengan gambar:



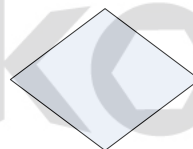
Gambar 3.10 Process

Database digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* adalah sebagai berikut:



Gambar 3.11 Database

Decision merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar dan salah. Operator logika ini digambarkan sebagai berikut:



Gambar 3.12 Decision

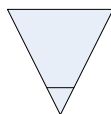
Manual input digunakan untuk melakukan proses input ke dalam *database* melalui keyboard. Manual input digambarkan dengan simbol:



Gambar 3.13 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. Off-line storage digambarkan dengan simbol:



Gambar 3.14 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. Bentuk simbol On-page reference adalah sebagai berikut:



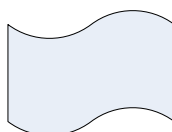
Gambar 3.15 On-Page Reference

Off-page reference memiliki sifat yang sedikit berbeda dengan On-page reference, karena simbol ini hanya digunakan apabila arus data yang ada dilanjutkan ke halaman yang berbeda. Bentuk simbolnya adalah:



Gambar 3.16 Off-Page Reference

Paper tape merupakan sebuah simbol yang umumnya menggantikan bentuk penggambaran jenis pembayaran yang digunakan (misal: uang) dalam transaksi yang ada pada sistem yang dirancang. Bentuk dari paper tape adalah dengan simbol:



Gambar 3.17 Paper Tape