

BAB III

LANDASAN TEORI

3.1 Konsep dasar Sistem Informasi

Konsep dasar dari Sistem Informasi terbagi atas dua pengertian. Yang pertama adalah sistem, dan yang kedua adalah sistem informasi itu sendiri.

3.1.1 Sistem

Definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya. (Herlambang, 2005)

3.1.2 Sistem Informasi

Data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, Informasi adalah data yang telah diolah dan

mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya. (Herlambang, 2005)

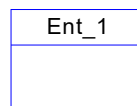
3.2 Distribusi

Distribusi adalah salah satu aspek dari pemasaran. Seorang atau sebuah perusahaan distributor adalah perantara yang menyalurkan produk dari pabrikan (*manufacturer*) ke pengecer (*retailer*). Setelah suatu produk dihasilkan oleh pabrik, produk tersebut dikirim (dan biasanya juga sekaligus dijual) ke suatu distributor. Distributor tersebut kemudian menjual produk tersebut ke pengecer atau pelanggan. (John Rawls, 1987).

3.3 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:

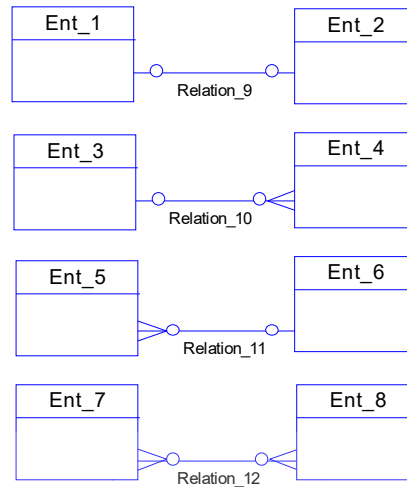
1. *Entity* atau entitas, digambarkan dalam bentuk persegi seperti pada gambar 3.1.



Gambar 3.1. *Simbol 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 3.2.



Gambar 3.2. *Simbol Relation of Entity*

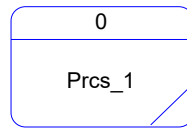
3.4 Data Flow Diagram

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 (*structured 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. *Simbol Process*

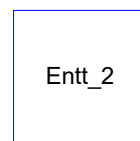
Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas proses. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa proses tersebut adalah sebuah *context diagram*. Diagram ini merupakan level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Pembuatan *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 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.

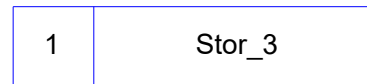
External entity disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku-pelaku sistem yang terkait, dapat berupa orang-orang, organisasi maupun instansi. *External entity* dapat memberikan masukan kepada *process* dan mendapatkan keluaran dari *process*. *External entity* digambarkan dalam bentuk sebagai berikut:



Gambar 3.4. *Simbol External Entity*

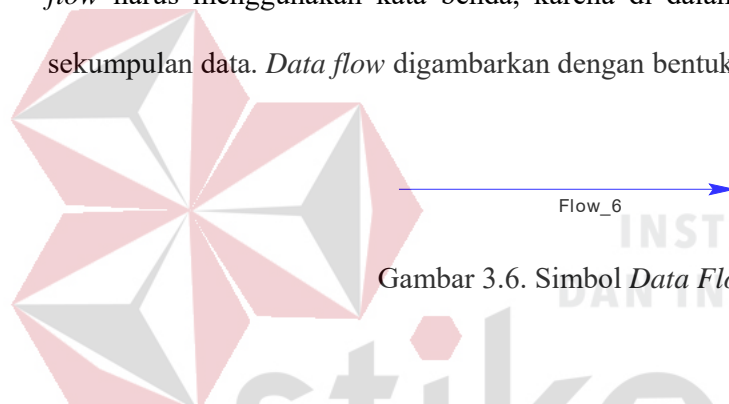
Data store digunakan sebagai media penyimpanan suatu data yang dapat berupa *file* atau *database*, arsip atau catatan manual, lemari *file*, dan tabel-tabel dalam *database*. Penamaan *data store* harus sesuai dengan bentuk data yang tersimpan pada *data store* tersebut, misalnya tabel pelamar, tabel pendidikan,

tabel lulus seleksi, dan lain-lain. *Data store* digambarkan dalam bentuk simbol sebagai berikut:



Gambar 3.5. Simbol *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. Simbol *Data Flow*

3.5 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).

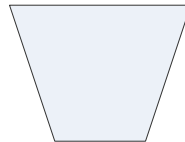
Terdapat berbagai macam bentuk simbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *decision*, *off-line storage*, *on-page reference*, dan *off-page reference*.

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:



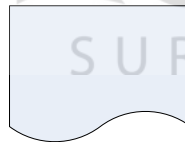
Gambar 3.7. *Simbol Terminator*

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



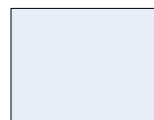
Gambar 3.8. *Simbol Manual Operation*

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:



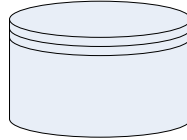
Gambar 3.9. *Simbol Document*

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



Gambar 3.10. *Simbol Process*

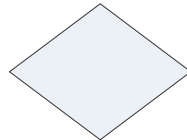
Database digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* adalah sebagai berikut:



Gambar 3.11. *Simbol 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:



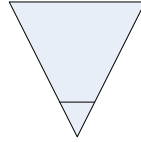
Gambar 3.12. *Simbol Decision*

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



Gambar 3.13. *Simbol 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. *Simbol 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:



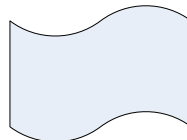
Gambar 3.15. *Simbol On-Page Reference*

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. *Simbol 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. *Simbol Paper Tape*

3.6 Visual Basic.NET

Cikal bakal bahasa pemrograman Visual Basic adalah *Beginer's All-purpose Symbolic Instruction Code* (BASIC) yang diciptakan sekitar tahun 1964 oleh Profesor John Kemeny dan Thomas Kurtz. Pada tahun 1975, Microsoft mengeluarkan satu produk pertamanya yaitu Microsoft Basic dan produk suksesnya adalah Quick Basic (Qbasic) yang banyak digunakan pada komputer pribadi atau *Personal Computer* (PC). Bahasa Pemrograman ini sangat populer karena mudah dipelajari dan bekerja di lingkungan grafis atau lebih dikenal dengan istilah *Graphical User Interface* (GUI).

Perkembangan Microsoft .NET yang selanjutnya pertama kali dipaparkan oleh Microsoft pada bulan Juli 2000 dalam *Professional Developers Conference* (PDC) di Orlando, Amerika Serikat. Kabarnya, setiap tahun Microsoft akan merilis Visual Studio .NET versi baru sebagai jawaban untuk menutup dan memperbaiki kekurangan yang ada dari versi-versi sebelumnya. (Yuswanto, 2006)

3.6.1 Visual Basic.NET 2008

Visual basic 2008 atau visual basic 9 merupakan paket teknologi bahasa pemrograman dari Visual Studio yang dikembangkan oleh Microsoft. Bahasa Visual Basic digunakan untuk membuat aplikasi Windows yang berbasis GUI (*Graphical User Interface*). GUI adalah sebuah aplikasi yang menampilkan antarmuka secara grafis. Hal ini memudahkan user/pemakai dalam mengoperasikan aplikasi.

Visual Basic merupakan salah satu bahasa *Object Oriented Programming* (OOP) dimana pemrograman difokuskan pada suatu objek tertentu. Selasin itu,

Visual Basic juga dikenal dengan sebutan *Even Driven Programming*, yaitu program akan bekerja setelah ada respon dari user berupa event/kejadian tertentu (seperti tombol diklik, menu dipilih, memasukkan data pada text field, dan lain-lain). Ketika sebuah event terdeteksi, kode yang berada pada objek yang mendapat event akan dijalankan.

Visual Basic 2008 mempunyai beberapa fitur baru dalam kompiler dan bahasa, IDE (*Integrated Development Environments*), akses data dan lain-lain. Fitur baru dalam dukungannya terhadap akses data/aplikasi berbasis database adalah sebagai berikut:

a. *Language-Integrated Query (LINQ)*

Language-Integrated Query (LINQ) adalah sebuah teknologi baru pada visual basic 2008 yang mendukung sintaks query dan bahasa baru lainnya.

LINQ membuat seolah-olah sebuah query database merupakan bagian sintaks /perintah Visual Basic. Hal ini menambah kemampuan query Visual Basic dengan perintah yang sederhana dan kemampuan yang tangguh saat bekerja dengan bermacam-macam data.

b. *Object Rational Designer (O/R Designer)*

Object Relational Designer atau disingkat dengan O/R Designer menyediakan desain visual untuk membuat kelas entity LINQ to SQL dan menghubungkannya berdasarkan onjek pada database. Fitur ini akan memandu para pengembang untuk membuat dean mengedit objek LINQ to SQL yang menghubungkan antara kode aplikasi dengan database

c. N-Tier

N-Tier mendukung penulisan datasets dengan menyediakan tingkatan (strata) lapisan *Dataset Designer* yang membuat lebih mudah untuk memisahkan Table Adapters dan menuliskan kode dataset ke dalam proyek terpisah. *N-Tier* juga disebut sebagai “aplikasi terdistribusi” dan “aplikasi multitier”. Aplikasi *N-tier* memisahkan pengolahan/pengaksesan data ke dalam tingkatan-tingkatan yang menyusun aplikasi secara terpisah yang didistribusikan antar klien dan server.

d. Peningkatan kemampuan hierarki

Hierarchical update capabilities (Peningkatan kemampuan hierarki) telah disatukan ke dalam *Dataset Designer*. Peningkatan ini menyediakan penciptaan kode yang menyimpan logika yang dibutuhkan untuk pemeliharaan integritas referensi antar tabel yang berkaitan di dalam suatu *database*.

e. *Local database caching*

Local database caching menyertakan database *SQL server Compact 3.5* ke dalam sebuah aplikasi dan mengatur sinkronisasi periodik data dengan *remote database* dengan *server*. *Local database caching* memungkinkan aplikasi mengurangi banyak roundtrips (keluar masuk data) antara aplikasi dengan *server database*, ketika data jarang berubah atau saat aplikasi berhubungan dengan *database server*. (Prabawati, 2009)

3.7 SQL Server

Microsoft SQL server 2005 adalah perangkat lunak pengolahan database yang sesuai untuk mengolah informasi dalam jumlah besar. Microsoft SQL server 2005 merupakan software manajemen database dimana database dapat didirikan sebagai kumpulan informasi atau data yang saling berhubungan yang mempunyai topik dan tujuan tertentu. Informasi atau data yang diolah tersebut disimpan sebuah file dengan ekstensi *.dbf.

SQL server 2005 adalah sebuah database rasional yang direncanakan untuk mendukung aplikasi dengan arsitektur client/server, sehingga memiliki arsitektur suatu integritas data yang tinggi dan dapat mengurangi lalu lintas *network*, karena hanya memberikan data yang diminta user saja. SQL server 2005 dapat mengandung beberapa database yang digunakan oleh beberapa user. (Amelia, 2007)

