

BAB III

LANDASAN TEORI

3.1 Konsep Dasar Sistem Informasi

3.1.1 Sistem

Menurut Herlambang (2005:116), 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.

3.1.2 Sistem Informasi

Menurut Herlambang (2005:121), 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.

3.2 Pembelian

Pembelian adalah usaha pengadaan barang-barang untuk perusahaan. Dalam perusahaan dagang pembelian dilakukan dengan dijual kembali tanpa mengadakan perubahan bentuk barang, sedangkan pada perusahaan manufaktur, pembelian dilakukan dengan merubah kembali bentuk barang.

Pembelian (*purchases*) adalah harga pembelian (harga pokok) barang dagang yang diperoleh perusahaan selama periode tertentu (Aliminsyah dan Padji, 2003:450).

Menurut Soemarso (1994:5) kegiatan pembelian dalam perusahaan dagang adalah:

1. Membeli barang dagang secara tunai atau kredit.
2. Membeli aktiva produksi untuk digunakan dalam kegiatan perusahaan.
3. Membeli barang dan jasa sehubungan dengan kegiatan perusahaan.

Sehingga dapat ditarik kesimpulan bahwa pembelian merupakan perkiraan yang digunakan untuk mencatat semua pembelian barang dagang dalam satu periode tertentu.

3.3 Analisa dan Perancangan Sistem

Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

Menurut Kendall (2003:7), Analisa dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

Berikut ini adalah proses dalam analisis dan perancangan sistem:

1. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta reasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. Relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*.

Menurut Marlinda (2004:28), *attribute* adalah kolom di sebuah relasi.

Macam-macam *attribute* yaitu:

a. *Simple Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya *entity* mahasiswa yang *attribute* nya NIM.

b. *Composite Attribute*

Attribute yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

c. *Single Value Attribute*

Attribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya Umur (tanggal lahir).

d. *Multi Value Attribute*

Multi value attribute adalah *attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute* nya pendidikan (SD, SMP, SMA).

e. *Null Value Attribute*

Null value attribute adalah *attribute* yang tidak memiliki nilai harga, misal *entity* tukang becak dengan *attribute* nya pendidikan (tanpa memiliki ijazah).

Entity Relationship Diagram ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*. Untuk itu *Entity Relationship Diagram* dibagi menjadi dua jenis model, yaitu:

a. *Conceptual Data Model (CDM)*

Conceptual Data Model adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

b. *Physical Data Model (PDM)*

Physical Data Model adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

2. Data Flow Diagram (DFD)

Data Flow Diagram atau yang sering disebut Bubble Chart atau diagram, model proses, diagram alur kerja atau model fungsi adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan aliran data baik secara manual maupun komputerisasi. DFD merupakan alat pembuatan model yang sering digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluaran. Untuk memudahkan proses pembacaan DFD, maka penggambaran DFD disusun berdasarkan tingkatan atau level dari atas ke bawah, yaitu :

1. *Context Diagram*

Merupakan diagram paling atas yang terdiri dari suatu proses dan menggambarkan ruang lingkup proses. Hal yang digambarkan dalam diagram konteks adalah hubungan terminator dengan sistem dan juga sistem dalam suatu proses. Sedangkan hal yang tidak digambarkan dalam *Context Diagram* adalah hubungan antar *terminator* dan *data source*.

2. *Diagram Zero (Level 0)*

Merupakan diagram yang berbeda diantara diagram konteks dan diagram detail serta menggambarkan proses utama dari DFD. Hal yang digambarkan dalam *Diagram Zero* adalah proses utama dari sistem serta hubungan *entity*, proses, alur data dan *data source*.

3. Diagram Detail (Primitif)

Merupakan penguraian dalam proses yang ada dalam *Diagram Zero*. Diagram yang paling rendah dan tidak dapat diuraikan lagi.

Data Flow Diagram (DFD) memiliki empat komponen, yaitu :

1. Terminator

Terminator atau *External Entity* atau kesatuan luar mewakili entitas *external* yang berkomunikasi dengan sistem yang dikembangkan. *Terminator* merupakan kesatuan di lingkungan sistem yang dapat berupa orang atau sistem yang berada di lingkungan luar sistem yang memberikan inputan maupun yang menerima output dari sistem serta berupa bagian atau divisi diluar sistem yang berkomunikasi dengan sistem. Terminator ini sering juga disebut entitas (*external*), sumber atau tujuan (*source and sink*).

2. Proses

Proses sering dikenal dengan *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input ke output, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu inputan atau lebih menjadi output dari sistem. Dilambangkan dengan lingkaran atau empat persegi panjang tegak dengan sudut tumpul. Proses diberi nama untuk menerangkan proses atau kegiatan apa yang sedang atau kegiatan yang akan dilaksanakan. Dan setiap proses harus diberi penjelasan lengkap sebagai berikut :

a. Identifikasi Proses

Umumnya berupa angka yang menunjukkan nomor dari proses dan ditulis pada bagian atas simbol.

b. Nama Proses

Menunjukkan apa yang sedang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya.

3. Data Store

Data Store digunakan sebagai sarana untuk pengumpulan data. *Data Store* disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada satu ujungnya atau dua garis horizontal. Suatu nama perlu diberikan pada *data store* menunjukkan nama dari filenya. *Data Source* biasanya berkaitan dengan penyimpanan *file* atau *database* yang dilakukan secara terkomputerisasi. *Data Store* dihubungkan dengan alur data hanya pada komponen proses pengertiannya sebagai berikut :

- a. Alur data dari store yang berarti sebagai pengaksesan data untuk suatu proses.
- b. Alur data ke proses berarti meng-*update* data seperti menambah data, mengurangi data maupun mengubah data.

4. Alur Data

Alur data dapat digambarkan dengan anak panah yang menuju ke dalam proses maupun ke luar proses. Alur data digunakan untuk menerangkan perpindahan data atau informasi dari suatu bagian ke bagian lainnya.

5. Syarat-syarat sebuah DFD :

- a. Pemberian nama untuk setiap komponen DFD.
- b. Pemberian nomor pada proses DFD.
- c. Penggambaran DFD serapi mungkin.

3.4 Konsep Dasar Basis Data

3.4.1 Database

Menurut Yuswanto (2005:2), database merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara database Relasional dan Non-Relasional. Pada database Non Relasional, sebuah database hanya merupakan sebuah file.

Menurut Marlinda (2004:1), database adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu database digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai), masalah keamanan (*security*), masalah integrasi, dan masalah data *independence* (kebebasan data).

3.4.2 Sistem Basis Data

Menurut Marlinda (2004:1), sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu perangkat keras (*hardware*), sistem operasi (*operating system*), basis data, sistem (aplikasi atau perangkat lunak) pengelola basis data (DBMS), pemakai (*user*), dan aplikasi (perangkat lunak) lain (bersifat opsional).

Keuntungan sistem basis data adalah:

1. Mengurangi kerangkapan data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga *update* dilakukan berulang-ulang.
2. Mencegah ketidakkonsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.
4. Integritas dapat dipertahankan.
5. Data dapat dipergunakan bersama-sama.
6. Menyediakan *recovery*.
7. Memudahkan penerapan standarisasi.
8. Data bersifat mandiri (*data independence*).
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat. Hal ini sangat erat hubungannya dengan pengontrolan kerangkapan data dan pemeliharaan keselarasan data.

Kerugian sistem basis data adalah:

1. Diperlukan tempat penyimpanan yang besar.
2. Diperlukan tenaga yang terampil dalam mengolah data.
3. Perangkat lunaknya mahal.
4. Kerusakan sistem basis data dapat mempengaruhi departemen yang terkait.

3.4.3 Database Management System

Menurut Marlinda (2004:6), *Database Management System* (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

Bahasa-bahasa yang terdapat dalam DBMS adalah:

1. *Data Definition Language* (DDL)

Pola skema basis data dispesifikasikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi perintah DDL adalah satu set tabel yang disimpan di dalam file khusus yang disebut *data dictionary/directory*.

2. *Data Manipulation Language* (DML)

Bahasa yang memperbolehkan pemakai mengakses atau memanipulasi data sebagai yang diorganisasikan sebelumnya model data yang tepat.

3. *Query*

Pernyataan yang diajukan untuk mengambil informasi. Merupakan bagian DML yang digunakan untuk pengambilan informasi.

DBMS memiliki fungsi sebagai berikut:

1. *Data Definition*

DBMS harus dapat mengolah pendefinisian data.

2. *Data Manipulation*

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

3. *Data Security dan Integrity*

DBMS dapat memeriksa *security* dan *integrity* data yang didefinisikan oleh DBA.

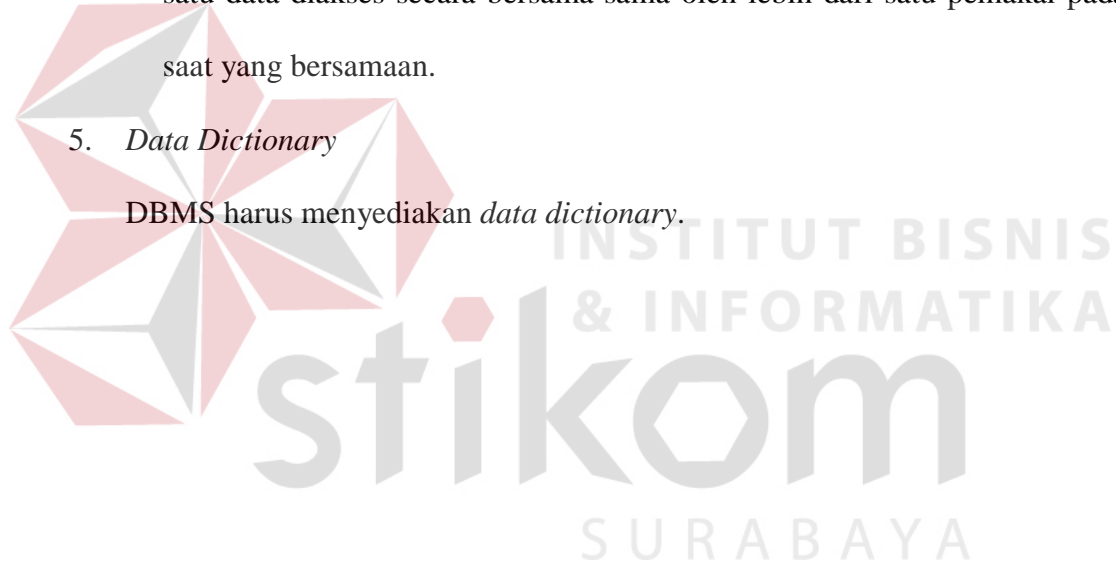
4. *Data Recovery dan Concurrency*

a. DBMS harus dapat menangani kegagalan-kegagalan pengaksesan basis data yang dapat disebabkan oleh kesalahan sistem, kerusakan disk, dan sebagainya.

b. DBMS harus dapat mengontrol pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat yang bersamaan.

5. *Data Dictionary*

DBMS harus menyediakan *data dictionary*.



3.5 Interaksi Manusia dan Komputer

Menurut Rizky (2006:4), Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-faktor utama dalam lingkungan interaksinya.

Deskripsi lain dari IMK adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif. Dikatakan juga bahwa sebuah desain antar muka yang ideal adalah yang mampu memberikan kepuasan terhadap manusia sebagai pengguna dengan faktor kapabilitas serta keterbatasan yang terdapat dalam sistem.

Pada implementasinya, IMK dipengaruhi berbagai faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan, antar muka, kendala, dan produktifitas.

3.6 .NET

.NET framework adalah suatu platform baru di dalam pemrograman untuk lingkungan yang terdistribusi luas (internet). Istilah .NET sering diasosiasikan dengan proses yang berjalan pada platform .NET.

Salah satu bentuk keunggulan dari platform ini terefleksi pada kompilasi sumber kode program, dimana semua sumber kode program akan dikompilasi menjadi *Microsoft Intermediate Language* (MSIL). Selanjutnya MSIL akan dikompilasikan oleh .NET Compiler menjadi bahasa mesin pada saat akan digunakan.

.NET merupakan alat untuk mewujudkan visi Microsoft pada jaringan internet dengan membentuk jaringan global yang saling berinteraksi agar dapat memberi pelayanan dan pertukaran data dengan cara yang lebih efisien dan terjamin dari segi keamanan.

Microsoft .NET Framework adalah produk software yang merupakan inti dari teknologi .NET. Produk ini bekerja secara terintegrasi dengan produk Microsoft lainnya, misalnya IIS. Ia terdiri dari beberapa modul seperti salah satu contohnya ASP.NET.

ASP.NET merupakan hasil pengembangan lebih lanjut dari ASP (*Active Server Page*), tetapi ia berbeda dari ASP, karena ASP.NET dibuat dengan dasar pemikiran yang berbeda sehingga program ASP tidak dapat dijalankan sebagai program ASP.NET. VBScript tidak lagi digunakan pada ASP.NET, sebagai gantinya anda dapat menggunakan VB.NET, C#, atau bahasa pemrograman lainnya. Penggunaan bahasa yang berbeda ini dimungkinkan karena ASP.NET mengadopsi konsep multi-language dalam pengembangan aplikasi program.

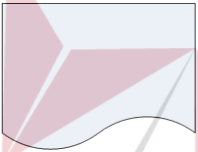



ADO.NET sebagai modul yang bekerja pada 'lingkungan terputus' (*disconnected fashion*), seperti pada layanan web adalah komponen kunci untuk mengakses sumber data (database) untuk memperoleh baris data atau memanipulasi database. Ia merupakan pengembangan lebih lanjut dari ADO (*ActiveX Data Objects*). Bersama dengan ASP.NET, ia memungkinkan terbentuknya halaman web yang dinamis (halaman web yang berubah-ubah tergantung pada inputan pengguna).

Pada dasarnya ASP.NET (.NET framework) adalah (dapat dipresentasikan) sebagai suatu tingkatan (*hierarchy*) Classes yang menyediakan

layanan dasar. Program ASP.NET mengandalkan penggunaan NameSpace sebagai API (*Application Program Interface*). NameSpace adalah skema penamaan untuk mengelompokkan tipe yang saling berhubungan.

3.7 Bagan Alir

Bagan alir (*flowchart*) merupakan bagan alir yang menunjukkan gambaran arus data dengan menggunakan simbol seperti pada tabel 3.1 berikut:

NO	SIMBOL	NAMA SIMBOL FLOWCHART	FUNGSI
1.		Dokumen	Untuk menunjukkan dokumen input dan output baik untuk proses manual, mekanik, atau komputer.
2.		Proses Komputerisasi	Menunjukkan kegiatan dari operasi program komputer.
3.		Database	Untuk menyimpan data.
4.		Penghubung	Menunjukkan hubungan di halaman yang sama.

5.		Penghubung Halaman Lain	Menunjukkan hubungan di halaman lain.
6.		Terminator	Menandakan awal /akhir dari suatu sistem.
7.		Decision	Menggambarkan logika keputusan dengan nilai <i>true</i> atau <i>false</i> .
8.		Input	Simbol inputan keyboard, menunjukkan data yang diinputkan melalui keyboard.
9.		Simpanan Offline	Untuk menunjukkan file non komputer yang diarsip urut angka .
10.		Catatan	Menunjukan data catatan

Tabel 3.1 : Flowchart