

## BAB III

### LANDASAN TEORI

#### 3.1 Konsep Dasar Sistem Inventory

Konsep dasar dari Sistem Inventori terbagi atas dua pengertian. Yang pertama adalah sistem, dan kemudian yang kedua adalah sistem inventori itu sendiri.

##### 3.1.1 Pengertian Sistem

Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sehingga sistem sangat diperlukan dalam memproses masukan (*input*), untuk menghasilkan informasi atau keluaran (*output*). Adapun keluaran yang didapatkan dapat berupa informasi yang dapat berfungsi sebagai dasar dalam pengambilan keputusan, baik oleh pimpinan atau siapa saja yang membutuhkan informasi tersebut (pihak-pihak yang berwenang dalam pengaksesan data).

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.

### 3.1.2 Pengertian Sistem Inventori

Menurut Koher dan Eric (2009:101), inventori adalah bahan baku dan penolong, barang jadi dan barang dalam proses produksi dan barang-barang yang tersedia, yang dimiliki dalam perjalanan dalam tempat penyimpanan atau konsinyasikan kepada pihak lain pada akhir periode. Sedangkan sistem informasi inventori adalah suatu sistem software yang akan membantu proses inventarisasi dengan menerapkan tertib administrasi inventori yang ketat. Pencatatan dari barang masuk, penyimpanan, sampai dengan barang keluar. Dilengkapi dengan sistem pelaporan yang sistematis dan akurat. Sistem informasi inventori ini dibuat dengan sistem *multi user* yang memungkinkan pengaksesan sistem informasi oleh beberapa user yang berbeda dalam suatu waktu.

### 3.2 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 sebuah 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*

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

Menurut Marlinda (2004 : 28), *Atribute* adalah kolom di sebuah relasi. Macam-macam atribut yaitu:

- a. *Simple Atribute*

*Atribut* ini merupakan *atribut* yang unik dan tidak dimiliki oleh *atribut* lainnya, misalnya entity mahasiswa yang atribut-nya NIM.

- b. *Composite Atribute*

*Composite Atribut* adalah *atribut* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

c. *Single Value Atribute*

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

d. *Multi Value Atribut*

*Multi Value Atribut* adalah *atribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *atribute*-nya pendidikan (SD, SMP, SMA).

e. *Null Value Atribut*

*Null Value Atribut* adalah *atribute* yang tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *atribute*-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*

*Conceptual Data Model* (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

b. *Physical Data Model*

*Physical Data Model* (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

c. *Data Flow Diagram*

Pada tahap ini, penggunaan notasi dapat membantu komunikasi dengan pemakai/*user* sistem untuk memahami sistem tersebut secara logika. Diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem ini dikenal dengan nama diagram arus data (*Data Flow Diagram*). DFD berfungsi untuk menggambarkan proses aliran data yang terjadi di dalam sistem dari tingkat yang tertinggi sampai yang terendah, yang memungkinkan untuk melakukan dekomposisi, mempartisi atau membagi sistem kedalam bagian-bagian yang lebih kecil dan yang lebih sederhana.

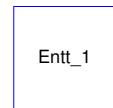
DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data tersebut (Kendall, 2003 : 241).

Simbol-simbol dasar dalam DFD antara lain:

a. *Eksternal Entity*

Suatu *Eksternal Entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.

Gambar 3.1 merupakan simbol entitas dalam DFD dalam model Gene dan Sarson.



Gambar 3.1 Simbol Eksternal Entity

b. *Data Flow*

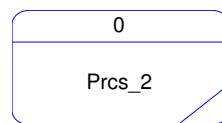
*Data Flow* atau aliran data disimbolkan dengan tanda panah. *Data Flow* menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Gambar 3.2 merupakan simbol *Data Flow*.



Gambar 3.2 Simbol Data flow

c. *Process*

Satu Proses dimana beberapa tindakan atau sekelompok tindakan dijalankan. Gambar 3.3 merupakan simbol Proses.



Gambar 3.3 Simbol Proses

d. *Data Store*

*Data Store* adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Gambar 3.4 merupakan simbol file penyimpanan/ *data store*.



Gambar 3.4 Simbol Data

### 3.3 Konsep Basis Data

Konsep dasar dari basis data terbagi atas tiga bagian, yaitu sistem basis *data*, *database*, dan *database management system*.

#### 3.3.1 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 (*database*), sistem (aplikasi atau perangkat lunak) pengelola basis data (DBMS), pemakai (*user*), dan aplikasi (perangkat lunak) lain (bersifat operasional).

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 ketidak konsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.
4. Integritas dapat dipertahankan.
5. Data dapat digunakan bersama-sama.
6. Menyediakan *recovery*.
7. Memudahkan penerapan standartisasi.
8. Data bersifat mandiri.
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat.

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.

### 3.3.2 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 standartisasi, *multiple user*, masalah keamanan, masalah integrasi, dan masalah *data independence*.

### 3.3.3 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 dispesifikasi 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 *Database Administrator* (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*.

### 3.4 Interaksi Masusia 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 implementasiannya, IMK dipengaruhi berbagai macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan, antar muka, kendala, dan produktifitas.

### 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).

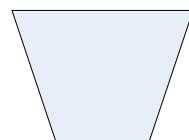
Terdapat berbagai macam bentuk simbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *decision*, *off-line storage*, *on-page reference*, dan *off-page reference*.

Terminator merupakan bentuk simbol yang digunakan sebagai tanda dimulainya jalan proses sistem ataupun tanda akhir dari sebuah pengeraaan suatu sistem.

Bentuk dari *terminator* adalah sebagai berikut:

Gambar 3.5. Simbol Terminator

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



Gambar 3.6 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:



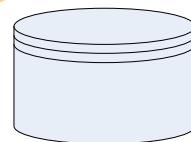
Gambar 3.7. Simbol Document

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



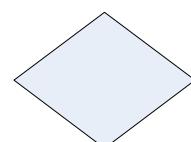
Gambar 3.8. Simbol Process

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



Gambar 3.9. 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:



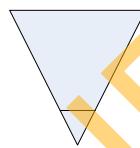
Gambar 3.10. Simbol Decision

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



Gambar 3.11. 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:



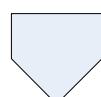
Gambar 3.12. Simbol Off-Line Storage

*On-page reference* digunakan sebagai simbol untuk menghubungkan bagian desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh dalam permasalahan letaknya. Bentuk simbol *On-page reference* adalah sebagai berikut:



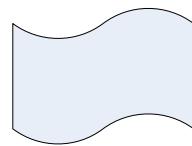
Gambar 3.13. Simbol On-Page Reference

*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 3.14. 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 3.15. Simbol Paper Tape

### 3.6 Power Designer

*Power designer* merupakan salah satu macam *software* yang digunakan untuk mengembangkan aliran data yang terjadi sehingga dapat dengan mudah diketahui alur yang akan digunakan. *Software* lain yang juga dapat digunakan adalah Microsoft Visio. Dengan Visio dapat dilakukan proses desain alur aliran data, tetapi hanya terbatas penggambaran saja dan tidak ada proses lanjutan.

Beberapa kelebihan yang dimiliki oleh *Power Designer* jika dibandingkan dengan software sejenis lainnya adalah adanya proses pengecekan terhadap kesalahan alur analisa aliran data dan juga dapat dilakukan proses *generate* untuk dilakukan proses selanjutnya.

*Power Designer* dapat digunakan untuk membuat berbagai macam diagram, pada umumnya adalah diagram aliran data (DFD) dan diagram relasi dari tabel entity (ERD). Proses lanjutan yang dapat di-generate adalah penjabaran aliran data menjadi lebih mendekil, serta merubah relasi antar tabel yang masih berupa konsep (CDM) menjadi ke berupa fisik (PDM).

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

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 di mana setiap kolom memiliki nama yang unik.

### 3.7 Visual Basic .Net

Visual Basic.Net merupakan salah satu produk untuk pengembang aplikasi dari Microsoft. Aplikasi yang dapat dikembangkan dengan Visual Basic.Net salah satunya adalah aplikasi database. Untuk aplikasi database ini, Visual Basic.Net mempunyai komponen pendukung, yaitu ADO.NET.

ADO.NET adalah teknologi akses data *universal* terbaru berdasarkan prinsip tanpa koneksi (*connectionless principle*) yang didesain untuk mempermudah batasan koneksi yang dahulunya harus diperhatikan ketika membuat aplikasi terdistribusi. Aplikasi hanya terhubung ke database untuk beberapa saat guna mengakses atau update data, kemudian diputus. Data yang diakses dapat disimpan pada salah satu objek ADO.NET, yaitu pada DataSet atau DataView. Keuntungan dari *disconnected architecture* ialah mampu menangani lebih banyak pengguna. Kelebihan lain data yang disimpan di DataSet berada di memori dan berformat XML.

Dalam ADO.NET tersedia beberapa provider yang dapat digunakan untuk mengakses suatu database, contohnya SQL Data Provider untuk mengakses database SQL Server, Oledb Data Provider untuk mengakses database Microsoft Access dan Oracle Data Provider untuk mengakses database Oracle.

STIKOM SURABAYA