

BAB II

LANDASAN TEORI

Dalam penyelesaian Tugas Akhir ini digunakan landasan teori yang membahas tentang teori yang dijadikan sebagai acuan dalam menyelesaikan permasalahan.

2.1 Sistem Informasi

Menurut Herlambang dan Tanuwijaya (2005:121), data sebenarnya merupakan fakta-fakta atau kejadian yang dapat berupa angka-angka atau berupa kode-kode tertentu. Data berupa angka atau wujud yang lain masih belum mempunyai arti atau kegunaan bagi penggunanya, sehingga harus diolah sedemikian rupa dengan menggunakan prosedur-prosedur tertentu untuk menghasilkan sebuah informasi bagi penggunanya. Secara singkat, informasi adalah data yang diolah dan mempunyai arti bagi penggunanya.

2.2 Data

Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti namun merupakan keterangan yang benar dan nyata. Data dapat diolah sehingga menghasilkan informasi.

Data adalah bentuk material atau bahan baku yang belum mempunyai makna atau belum berpengaruh langsung kepada pengguna hingga perlu diolah untuk menghasilkan suatu yang lebih bermakna. (Agus Mulyanto, 2009:15)

2.3 Klinik

Menurut PERMENKES RI Nomor 9 Tahun 2014 Bab 1 Pasal 1 menyatakan Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan atau spesialistik.

2.4 Aplikasi

Aplikasi (*Application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Dapat disimpulkan bahwa aplikasi merupakan *Software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan, penambahan data (Dhanta (2009:32)).

2.5 Inventory

Menurut Richardus (2003:3), inventori adalah barang-barang yang biasanya dapat dijumpai di gudang tertutup, lapangan, gudang terbuka, atau tempat-tempat penyimpanan lainnya, baik berupa bahan baku, barang setengah jadi, barang jadi, barang-barang untuk keperluan operasi, atau barang-barang untuk keperluan suatu proyek.

Jenis-jenis inventori akan berbeda sesuai dengan bidang atau kegiatan normal usaha perusahaan tersebut. Berdasarkan bidang usaha perusahaan dapat terbentuk perusahaan jasa, perusahaan dagang, ataupun perusahaan industry (*manufacture*). Untuk dapat memahami perbedaan dan keberadaan dari tiap-tiap

jenis persediaan tersebut maka dapat dilihat dari penggolongan inventori secara garis besar, menurut Richardus (2003:5), yaitu:

1. Inventori bahan baku (*raw material*), merupakan barang-barang yang diperoleh untuk digunakan dalam proses.
2. Inventori bahan dalam proses (*goods in process*), yang juga disebutkan sebagai pekerjaan dalam proses (*work in process*) terdiri dari bahan baku yang sebagian telah diproses dan perlu dikerjakan lebih lanjut.
3. Inventori barang jadi (*finished goods*), merupakan produk/barang yang telah selesai diproduksi dan menjadi persediaan perusahaan untuk dijual atau digunakan untuk operasi perusahaan.

2.6 Economic Order Quantity (EOQ)

Salah satu keputusan yang terpenting dalam mengatur persediaan adalah berapa banyak barang yang akan di pesan kepada supplier. Dengan EOQ dapat dihitung berapa sebaiknya pesanan dilakukan dengan asumsi yang dapat diketahui secara pasti. (Aquilani 2001:517)

EOQ dapat dihitung dengan menghubungkan antara biaya penyimpanan per unit, biaya pemesanan setiap kali pesan, jumlah kebutuhan bahan baku untuk satu periode dan harga beli barang per unit.

Rumus EOQ :

$$EOQ = \sqrt{\frac{2(Annual\ Usage)(Order\ Cost)}{(Annual\ Carrying\ cost\ per\ unit)}}$$

atau

$$EOQ = \sqrt{\frac{2DS}{H}}$$

dimana :

D = Kuantitas penggunaan per periode

S = Biaya Per Pemesanan

H = Biaya penyimpanan per unit per periode.

ROP (Reorder Point) Menurut Sofjan Assauri (2004;196), tingkat pemesanan kembali (*reorder point*) adalah : “Tingkat pemesanan kembali adalah suatu titik atau batas dari jumlah persediaan yang ada pada suatu saat dimana pemesanan harus diadakan kembali”.

Faktor-faktor yang mempengaruhi titik pemesanan kembali adalah :

- Lead Time. Lead time adalah waktu yang dibutuhkan antara barang yang dipesan hingga sampai diperusahaan.
- Tingkat pemakaian bahan baku rata-rata persatuan waktu tertentu.
- Persediaan Pengaman (*Safety Stock*), yaitu jumlah persediaan barang minimum yang harus dimiliki oleh perusahaan untuk menjaga kemungkinan keterlambatan datangnya bahan baku.

Dari ketiga faktor di atas, maka reorder point dapat dicari dengan rumus berikut :

Rumus ROP :

$$ROP = (D \times L) + S$$

dimana :

D = Penggunaan rata-rata per periode







L = Lead time (waktu tunggu)

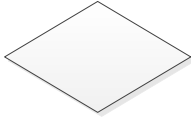
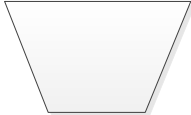
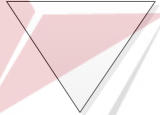
S = Safety stock (penggunaan cadangan)

2.7 Document Flow

Document Flow adalah bagan yang menunjukkan alur dalam program ataupun prosedur sistem secara fisik. Bagian alur digunakan terutama untuk alat bantu komunikasi dan dokumentasi. . (Jogiyanto, 2005 : 701) Bagan alur sistem digambar dengan menggunakan simbol – simbol antara lain sebagai berikut :

Tabel 2.1 Simbol Data Flow Diagram

No.	Simbol	Nama Simbol <i>Flowchart</i>	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.

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
7.		Decision	Menggambarkan logika keputusan dengan nilai <i>true</i> atau <i>false</i> .
8.		Kegiatan Manual	Untuk menunjukkan pekerjaan yang dilakukan secara manual.
9.		Simpanan Offline	Untuk menunjukkan file non-komputer yang diarsipurut angka.

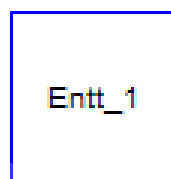
2.8 Data Flow Diagram

Menurut (Kendall 2003: 241), *Data Flow Diagram* menggambarkan pandangan sejauh mungkin mengenai masukan, proses dan keluaran sistem, yang berhubungan dengan masukan, proses, dan keluaran dari model sistem yang dibahas. Serangkaian diagram aliran data berlapis juga bisa digunakan untuk merepresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem. Prosedur-prosedur tersebut yaitu konseptualisasi bagaimana data-data berpindah di dalam organisasi, proses-proses atau transformasi dimana data-data melalui, dan apa keluarannya. Jadi, melalui suatu teknik analisa data terstruktur yang disebut *Data Flow Diagram*, penganalisis sistem dapat merepresentasi proses-

proses data di dalam organisasi. Menurut (Kendall 2003: 265), dalam memetakan *Data Flow Diagram*, terdapat beberapa simbol yang digunakan antara lain:

1. *External entity*

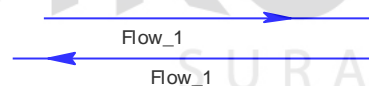
Suatu *external entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.



Gambar 2.1 Simbol *External Entity*

2. *Data Flow*

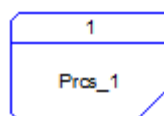
Data Flow atau aliran data disimbolkan dengan data tanda panah. Aliran data menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses.



Gambar 2.2 Simbol *Data Flow*

3. *Process*

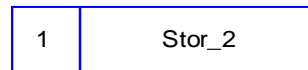
Suatu proses dimana beberapa tindakan atau sekelompok tindakan dijalankan.



Gambar 2.3 Simbol *Process*

4. Data Store

Data store adalah simbol yang digunakan untuk melambangkan proses penyimpanan data.



Gambar 2.4 Simbol *Data Store*

2.9 Entity Relations Diagram (ERD)

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *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. *Attribute* yaitu uraian dari entitas dimana mereka dihubungkan atau dapat dikatakan sebagai *identifier* atau *descriptors* dari entitas.

Entitas digolongkan menjadi *independent* atau *dependent entity*. *Independent entity* adalah apa yang tidak bersandar pada yang lain sebagai identifikasi. Suatu *dependent entity* adalah apa yang bersandar pada yang lain sebagai identifikasi. Selain digolongkan menjadi *independent* atau *dependent entity*, terdapat jenis- jenis entitas khusus yaitu:

1. *Associative Entity*

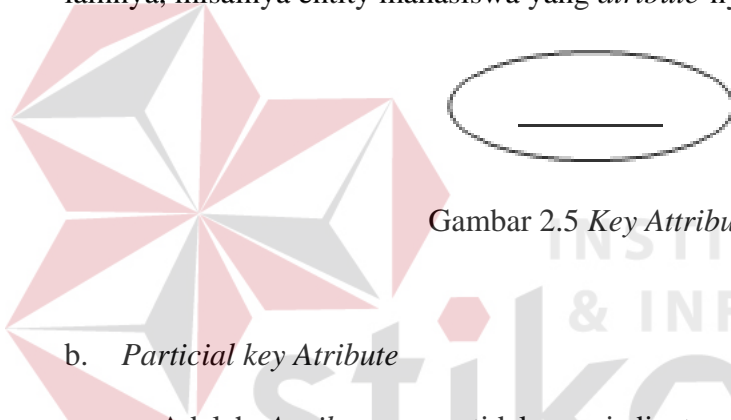
Associative Entity (juga dikenal sebagai *intersection entity*) adalah entitas yang *digunakan* oleh rekanan dua entitas atau lebih untuk menyatukan suatu hubungan banyak - ke - banyak (*Many to Many*).

2. *Subtypes Entity*

Subtypes Entity digunakan di dalam hierarki generalisasi (*generalization hierarchies*) untuk menyajikan suatu subset kejadian dari entitas orang tua, yang disebut *supertype*, tetapi yang memiliki atribut atau hubungan yang berlaku hanya untuk *subset*. Menurut (Marlinda 2004: 28), *attribute* sebagai kolom di sebuah relasi mempunyai macam-macam jenis *attribute* yaitu :

a. *Key Attribute*

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



Gambar 2.5 *Key Attribute*

b. *Particual key Attribute*

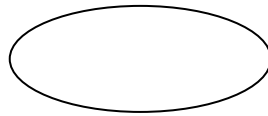
Adalah *Attribute* yang tidak menjadi atau merupakan anggota dari *Key Primer*. Misalnya antara Cabang (toko) dan kode cabang.



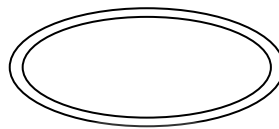
Gambar 2.6 *Particual Key Attribute*

c. *Single Vallue Attribute*

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

Gambar 2.7 *Single Value Attribute*d. *Multi Value Attribute*

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

Gambar 2.8 *Multi Value Attribute*e. *Composite Attribute*

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

Gambar 2.9 *Composite Attribute*f. *Derived Attribute*

Attribute yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari table *Attribute* atau table lain yang berhubungan.

Gambar 2.10 *Derived Attribute*

Model *Entity - Relationship* (ER) mula-mula diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entities*) dan hubungan (*relationship*).

Komponen dasar model merupakan diagram *entity-relationship* yang digunakan untuk menyajikan objek data secara *visual*. *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:

1. *Conceptual Data model*

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

2. *Physical Data Model*

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

2.10 SQL Server 2008

Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Hidayatullah, 2012). Sistem manajemen basis data relasional (RDBMS) adalah sebuah program komputer yang dirancang untuk mengatur atau memanajemen sebuah basis data sebagai sekumpulan data yang disimpan secara teratur, dan melakukan operasi-

operasi data atas permintaan penggunanya. Beberapa contoh RDBMS yang sering digunakan adalah Oracle, SQL Server, Microsoft Access, dan MySQL. SQL adalah singkatan dari *Structured Query Language*. Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft.

Bahasa Query 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 menengah, akan tetapi dengan seiring perkembangannya SQL Server digunakan pada basis data berskala besar. Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (Tabular Data Stream). Selain 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.

2.11 Microsoft Visual Basic .Net

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem *.NET Framework*, dengan menggunakan bahasa *BASIC*. Dengan menggunakan alat ini, para pembuat program dapat membangun aplikasi *Windows Forms*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*,

Visual C#, atau *visual j#*) atau juga dapat diperoleh secara terpadu dalam *Microsoft visual Studion .NET* (Yuwanto, 2005).

Bahasa *Visual Basic .NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsot Visual Basic* versi sebelumnya yang dimplementasikan diatas *.NET Framework*. Peluncurannya mengundang kontrovensi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.

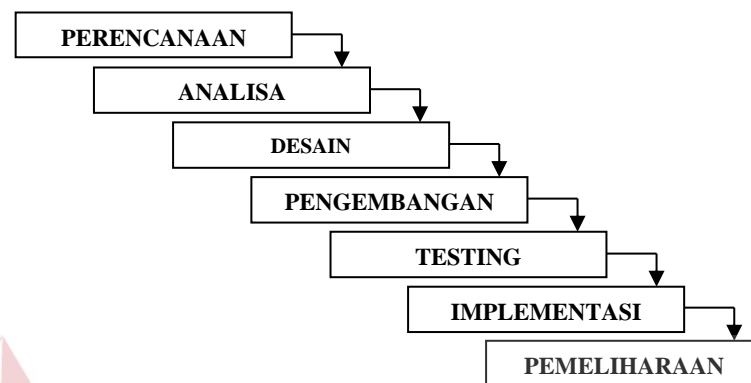
2.12 System Development Life Cycle

System development life cycle (SDLC) adalah keseluruhan proses dalam membangun sistem informasi melalui beberapa tahapan. Ada beberapa model SDLC namun yang apling sering digunakan dan paling populer adalah model *waterfall*. Model ini disebut *waterfall* karena dikerjakan langkah per langkah seperti air mengalir. Adapun model lain dari SDLC yaitu *fountain*, *spiral*, *rapid prototyping*, *incremental*, *build & fix*, *System development life cycle* dan *synchronize & stabilize*.

Dengan menggunakan SDLC maka proses membangun sebuah sistem informasi dibagi menjadi beberapa tahapan dan dikerjakan oleh tim yang berpengalaman dalam bidang tersebut. Biasanya pemula mengabaikan SDLC dan berfokus pada pemrograman sehingga sistem informasi yang dihasilkan tidak bermutu karena tanpa adanya perencanaan, desain, dan konsep.

System development life cycle adalah keseluruhan proses dalam membangun system melalui beberapa tahapan. Model *system development life*

cycle yang paling banyak digunakan oleh sistem analis dan *programmer* adalah model *waterfall* (Hartono, 2004). Tahapan dalam membangun sistem informasi menggunakan model *waterfall* dapat dilihat pada Gambar 2.11.



Gambar 2.11 Tahapan Membangun Sistem (Hartono, 2004)

1. Perencanaan

Tahap perencanaan adalah membuat semua rencana yang berkaitan dengan proyek sistem informasi. Tahap perencanaan merupakan proses dasar untuk memahami mengapa sebuah sistem harus dibangun. Pada tahap ini diperlukan analisa kelayakan dengan mencari data kepada narasumber.

2. Analisa

Tahap analisa adalah melihat proses bisnis pada perusahaan yang ada saat ini. Tahap analisa bertujuan untuk mendapatkan jawaban dari penggunaan sistem dan cara kerja sistem dan dari tahap analisa ini yang akan didapatkan cara untuk membangun sebuah sistem yang baru.

3. Desain

Tahap desain adalah langkah yang paling penting karena tahap ini yang menentukan berjalan atau tidaknya sebuah aplikasi. Tahapan desain meliputi desain *database*, desain *interface*, desain *report*.

4. Pengembangan

Tahap pengembangan adalah tahap menulis sebuah *code-code* dengan menggunakan Bahasa pemrograman tertentu sehingga akan menghasilkan sebuah aplikasi. Penulisan *code-code* berdasarkan algoritma dan logika sesuai dengan kebutuhan sistem.

5. Testing

Tahap testing merupakan tahap yang digunakan untuk mengidentifikasi apakah dalam sebuah sistem yang dibuat ada ketidaksesuaian dengan apa yang diharapkan. Testing dilakukan agar dapat diketahui apakah telah sesuai dengan kebutuhan sistem sebelum sistem benar-benar digunakan.

6. Implementasi

Tahap implementasi merupakan tahap untuk menerapkan sebuah sistem informasi yang telah dibangun oleh pengembangan agar user dapat menggantikan proses bisnis yang lama. Dalam tahap ini, user dilatih agar dapat menjalankan sistem yang baru.

7. Pemeliharaan

Tahap pemeliharaan adalah upaya untuk, memperbaiki, menjaga, menanggulangi serta mengembangkan sistem. Pemeliharaan ini dilakukan untuk menjaga kinerja sistem yang telah dibuat dapat berjalan dengan baik dan dapat digunakan secara optimal.