

BAB III

LANDASAN TEORI

3.1 Sistem Informasi

Ialah sebuah set elemen atau komponen terhubung satu sama lain yang mengumpulkan (*input*), memanipulasi (*process*), menyimpan, dan menghasilkan (*output*) data dan informasi, dan memberikan pembetulan (mekanisme *feedback*) untuk memenuhi tujuan. Mekanisme *feedback* ialah komponen yang membantu organisasi meraih tujuan, seperti meningkatkan profit atau meningkatkan *customer service* (Stair, 2010).

3.2 System Life Cycle Development (SDLC)

Menurut Turban (2003), *System Development Life Cycle* (SDLC) atau Siklus Hidup Pengembangan Sistem adalah metode pengembangan sistem tradisional yang digunakan sebagian besar organisasi saat ini. SDLC adalah kerangka kerja (*framework*) yang terstruktur yang berisi proses-proses sekuensial di mana sistem informasi dikembangkan. Salah satu tipe SDLC yang paling awal dan paling banyak digunakan adalah metode *Waterfall*. *Waterfall methode* sering dianggap sebagai pendekatan klasik dengan siklus hidup pengembangan sistem. Pembangunan dengan metode *Waterfall* memiliki tujuan yang berbeda untuk setiap fase pembangunan. Setelah fase pembangunan selesai, hasil pengembangan ke tahap berikutnya dan tidak ada jalan kembali. Adapun tahapan-tahapan pada metode *Waterfall* antara lain :

1. *Requirement Analysis*

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

2. *System Design*

Tahap ini dilakukan sebelum melakukan *coding*. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasi kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum. Dalam tahap ini lebih mengarah pada bagian uji fungsi.

4. *Integration & Testing*

Pada tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak. Hal ini berkaitan dengan uji data.

3.3 Manajemen Kearsipan

Adalah pekerjaan pengurusan arsip yang meliputi pencatatan, pengendalian, dan pendistribusian, penyimpanan, pemeliharaan, pengawasan, pemindahan, dan pemusnahan (Alimsyah, 2005).

3.4 Analisa Dan Perancangan Sistem

Analisa sistem ialah sebuah tahap yang paling penting dimana merupakan tahap awal untuk mengevaluasi permasalahan yang terjadi serta kendala-kendala yang dihadapi sehingga didapatkan point-point penting berupa kebutuhan *User* (dalam hal ini ialah perusahaan) untuk pendesainan dan implementasi program nantinya. Hasil analisa dan perancangan sistem nantinya juga digunakan sebagai bahan acuan dalam melakukan hasil evaluasi apakah aplikasi yang sudah di desain dan di implementasi sudah memenuhi kebutuhan dari *user* atau belum.

Analisa yang efektif akan memudahkan pekerjaan penyusunan rencana yang baik ditahap berikutnya. Sebaliknya, kesalahan yang terjadi pada tahap analisa ini akan menyebabkan kesulitan yang lebih besar, bahkan dapat menyebabkan penyusunan sistem gagal (Jogiyanto, 2005).

Untuk itu diperlukan ketelitian di dalam menganalisa sehingga tidak terdapat kesalahan dalam tahap selanjutnya, yaitu tahap perancangan sistem. Langkah-langkah yang diperlukan didalam menganalisa sistem adalah :

- a. Tahap perencanaan sistem
- b. Tahap analisa sistem
- c. Tahap perancangan sistem
- d. Tahap penerapan sistem
- e. Membuat laporan dan hasil analisa

3.5 Document Flow

Document Flow menggambarkan hubungan antara *input*, *process*, dan *output*. Sebuah *Document Flow* akan mengidentifikasi input yang masuk ke dalam sistem dan asal dari input tersebut. Input dapat berupa data baru yang masuk ke dalam sistem atau data yang disimpan untuk digunakan di masa yang akan datang. *Document Flow* juga menampilkan logika yang digunakan komputer ketika melakukan proses dalam sistem. Hasil informasi baru merupakan komponen output yang dapat disimpan untuk digunakan di masa yang akan datang dan ditampilkan dalam layar komputer atau dicetak kertas. Dalam beberapa hal, output dari sebuah proses adalah input untuk proses lainnya (Romney & Steinbart, 2000)

Berikut simbol-simbol yang digunakan dalam Dokumen Flow :

Tabel 3.1 Simbol-Simbol dalam *Document Flow*

No	Simbol	Nama	Keterangan
1		Dokumen	Dokumen yang dioleh dengan cara manual yang dicetak
2		Data	Sebuah data yang akan dijadikan sebuah dokumen maupun sebuah output
3		Manual Input	Input yang dilakukan oleh user dari keyboard maupun Device lainnya

3.6 Data Flow Diagram

Data Flow Diagram ialah representasi yang berbentuk diagram dari sebuah sistem, yang menyediakan gambaran sistem yang akan dibuat secara komplit, jelas, dan mudah dimengerti (ISRD, 2007). DFD sendiri terdiri atas :

Tabel 3.2 Simbol-Simbol pada DFD

Nama	Simbol	Deskripsi
<i>Terminator</i>		<i>External Entity</i> , memberikan input maupun menerima output dari sistem
<i>Proses</i>		Aktifitas pengolahan <i>input</i> (masukan) menjadi <i>output</i> (Keluaran)
<i>Data Flow</i>		Aliran data pada sistem yang sifatnya : - Antar proses - Antar <i>terminator</i> & proses - Antar proses & <i>data store</i>
<i>Data Store</i>		penyimpanan data pada <i>database</i> (berbentuk tabel)

Dalam DFD dibagi menjadi beberapa bagian yaitu :

1. Diagram Konteks (*Context Diagram*) alat pembuatan model yang memungkinkan menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain via aliran data
2. DFD Level 0 Satu lingkaran sistem besar yang mewakili lingkaran-lingkaran (sistem) yang ada di dalamnya
3. DVD Level 1 dst rincian lingkaran-lingkaran kecil yang menguraikan proses apa saja yang terdapat pada DVD level 0 ataupun sebelumnya (DFD lv 1)

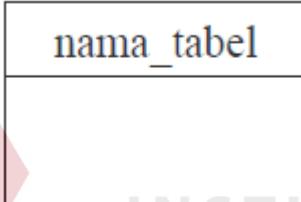
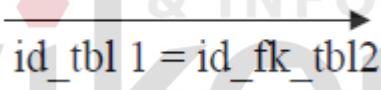
3.7 Conceptual Data Model

Conceptual Data Model (CDM) adalah 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. Biasanya CDM direpresentasikan dalam bentuk *Entity Relationship Diagram* (Ramadhani, 2013). Adapun manfaat penggunaan CDM dalam perancangan *database* :

1. Memberikan gambaran yang lengkap dari struktur basis data yaitu arti, hubungan, dan batasan-batasan
2. Alat komunikasi antar pemakai basis data, *designer*, dan analis.

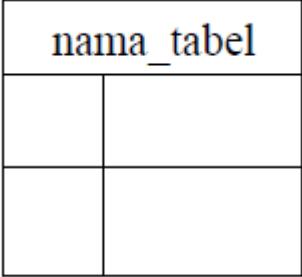
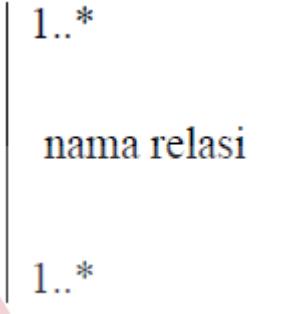
Tabel 3.3 Simbol Conceptual Data Model

Nama	Simbol	Deskripsi
<i>Tabel</i>		Tabel yang menyimpan data dalam basis data
<i>Relasi</i>		Relasi antar tabel

3.8 Physical Data Model

PDM Merupakan 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. (Ramadhani, 2013)

Tabel 3.4 Simbol Physical Data Model

Nama	Simbol	Deskripsi
<i>Tabel</i>		Entitas atau tabel yang menyimpan data dalam basis data
<i>Relasi</i>		Relasi antar tabel yang terdiri atas nama relasi dan multiplicity

