

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Jogiyanto, 2005:1).

Istilah sistem secara umum dapat didefinisikan sebagai kumpulan hal atau elemen yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. Sistem mempunyai karakteristik atau sifat – sifat tertentu, yaitu : Komponen Sistem, Batasan Sistem, Lingkungan Luar Sistem, Penghubung Sistem, Masukan Sistem, Keluaran Sistem, Pengolahan Sistem dan Sasaran Sistem (Sutanta, 2009:4).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. (Jogiyanto, 2005:11).

Informasi adalah data yang diolah menjadi bentuk yang berguna dan menjadi berarti bagi penerimanya. Kegunaan informasi adalah untuk mengurangi ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut. Kualitas informasi sangat

dipengaruhi atau ditentukan oleh beberapa hal yaitu : Relevan (*Relevancy*), Akurat (*Accuracy*), Tepat waktu (*Timeliness*), Ekonomis (*Economy*), Efisien (*Efficiency*), Ketersediaan (*Availability*), Dapat dipercaya (*Reliability*), Konsisten (Sutanta, 2009:8).

Sistem informasi berdasarkan Jogiyanto (2005:18) adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.2 *System Development Life Cycle (SDLC)*

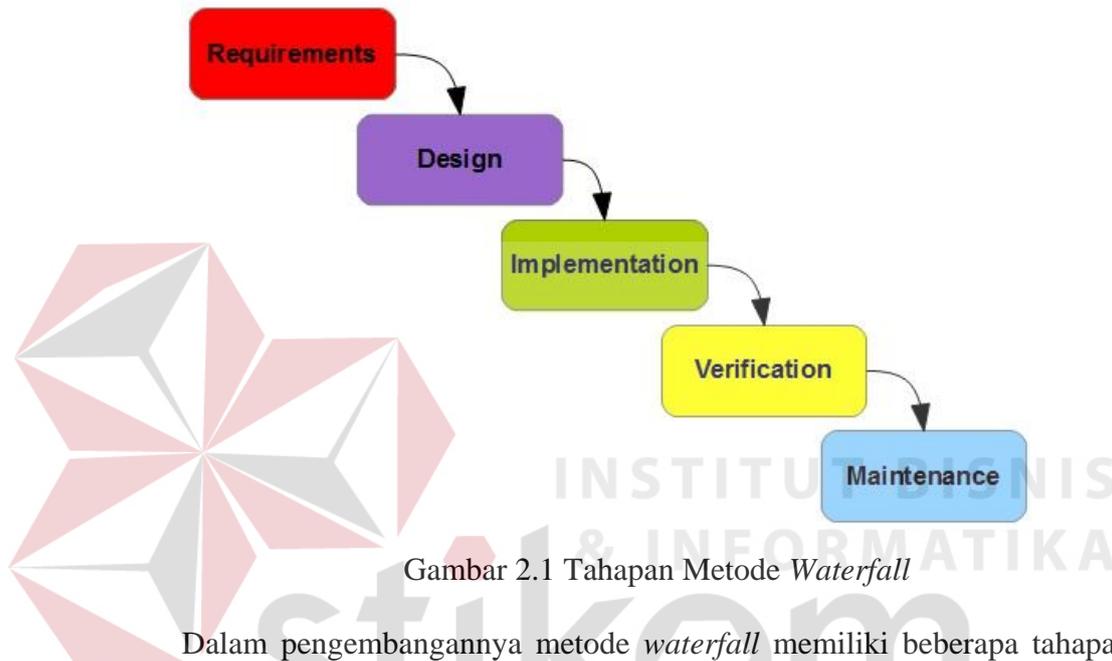
System Development Life Cycle (SDLC) adalah sebuah metode yang digunakan untuk mengembangkan sebuah sistem. SDLC adalah sebuah proses logika yang digunakan oleh seorang sistem analist untuk mengembangkan sebuah sistem informasi yang melibatkan *requirements, validation, training* dan pemilik sistem (Mulyani, 2009:32).

Menurut Phillip & Seppo (2012) SDLC merupakan sebuah kerangka kerja sistematis yang tersusun berdasar penentuan spesifikasi, pembuatan desain, *programming, testing* dan pemeliharaan *software*.

Sementara berdasarkan Turban (2003) SDLC merupakan sebuah kerangka kerja metode pengembangan sistem yang *sequencial*.

Ada beberapa model SDLC, tetapi dalam hal ini metode yang digunakan adalah metode *waterfall*. Penyusun memilih model *waterfall*, karena langkah – langkahnya berurutan dan sistematis.

Menurut Pressman (2001) Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Tahapan metode *waterfall* dapat di gambarkan sebagai berikut :



Gambar 2.1 Tahapan Metode *Waterfall*

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang runtut: *requirement* (analisis kebutuhan), *system design* (desain sistem), *Coding & Testing*, Penerapan Program, pemeliharaan. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut :

1. *Requirement Analysis* (Analisa Kebutuhan)

Pada tahap ini pengembang sistem diperlukan suatu komunikasi yang bertujuan untuk memahami *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, *survey* atau diskusi. Informasi tersebut dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *System Design* (Desain Sistem)

Spesifikasi kebutuhan dari tahap pertama akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation* (Implementasi)

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap berikutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai Unit Testing.

4. *Integration & Testing* (Uji Coba)

Semua unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian masing-masing unit. Pasca integrasi seluruh sistem diuji untuk mengecek setiap kesalahan dan kegagalan.

5. *Operation & Maintenance* (Pemeliharaan)

Ini merupakan tahap terakhir dalam model *waterfall*. *Software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unitsistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.3 Pengertian Program

Pada umumnya program adalah sederetan instruksi atau statement yang tentunya dalam bahasa yang dimengerti oleh komputer. Instruksi tersebut

berfungsi untuk mengatur pekerjaan apa saja yang akan dilakukan oleh komputer agar mendapatkan dan menghasilkan suatu hasil atau keluaran yang diharapkan.

Menurut Binanto (2005:1) kata program dapat diartikan:

1. Untuk mendeskripsikan instruksi-instruksi tersendiri, yang biasanya disebut *source code*, yang dibuat *programmer*.
2. Untuk mendeskripsikan suatu keseluruhan bagian dari *software* yang *executable*.

Dapat juga dikatakan bahwa sebuah program merupakan himpunan atau kumpulan instruksi tertulis yang dibuat oleh *programmer* atau suatu bagian *executable* dari sebuah *software*.

Pengertian Pemrograman menurut Jogiyanto (2005:582) merupakan “kegiatan menulis kode program yang akan dieksekusi oleh komputer”.

Menurut Indrajani (2007:22), bahasa pemrograman adalah “perangkat lunak atau *software* yang dapat digunakan dalam proses pembuatan program yang melalui beberapa tahapan-tahapan penyelesaian masalah”.

Proses pemrograman komputer bukan saja sekedar menulis suatu urutan instruksi yang harus dikerjakan oleh komputer akan tetapi bertujuan untuk memecahkan suatu masalah serta membuat mudah pekerjaan pengguna komputer (*user*). Didalam membuat sebuah program komputer, tentu tidak terlepas dari sifat individu pemrogram (*Programmer*). Karakteristik seorang pemrogram yang mutlak dimiliki yaitu:

- a. Memiliki pola pikir yang logis
- b. Memiliki ketekunan dan ketelitian yang tinggi
- c. Memiliki penguasaan bahasa pemrograman yang baik

- d. Memiliki pengetahuan teknik pemrograman yang baik

Untuk membuat suatu Program yang kompleks tahap-tahap yang harus dilakukan *programmer* adalah :

1. Definisi Masalah

Programmer harus memahami permasalahan yang timbul kemudian mengidentifikasi permasalahan yang ada sehingga dapat menentukan batasan masalah.

2. Analisa Kebutuhan

Programmer harus menentukan kebutuhan data untuk masukan dan keluaran yang diminta, bahasa pemrograman yang digunakan serta tipe komputer apa sebagai pendukungnya.

3. Desain Algoritma

Algoritma yang didesain harus memiliki kebenaran secara logika sebelum siap diimplementasikan ke dalam bentuk program.

4. Bahasa Pemrograman

Bahasa Pemrograman adalah media untuk membuat Program.

5. *Testing* dan *Debugging*

Testing untuk menguji program sampai terbebas dari kesalahan.

Debugging untuk mengoreksi kesalahan yang terdeteksi.

Ada tiga macam kesalahan yang biasanya terjadi :

- a. *Syntax Error*

Bentuk kesalahan program yang terjadi karena kesalahan dalam hal penulisan instruksi di dalam program.

b. *Run Time Error*

Bentuk kesalahan program yang terjadi karena adanya proses aritmatik yang tidak dapat diproses.

c. *Logical Error*

Bentuk kesalahan yang terjadi karena kesalahan logika program yang dibuat oleh *programmer*.

6. Dokumentasi digunakan untuk *file* cadangan (*Backup*)

7. Pemeliharaan

Yaitu dalam upaya menghindari kerusakan atau hilangnya suatu program baik dari *Factor Software* (*Virus Program*) ataupun dari *Factor Brainware* (*Human Error*).

2.4 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 (kesatuan), dan masalah data *independence* (kebebasan data).

2.5 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.

Berikut ini merupakan Bahasa-bahasa yang ada dalam DBMS, yakni:

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.

Fungsi-fungsi dari DBMS terdiri dari 3 yaitu *Data Definition*, *Data Manipulation*, *Data Security* dan *Integrity*. Berikut adalah penjelasan dari ketiga fungsi tersebut:

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*.

2.6 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 macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan, antar muka, kendala dan produktifitas.

