

BAB II LANDASAN TEORI

2.1. Konsep Dasar Sistem

2.1.1. Pengertian Sistem

Sistem adalah jaringan dari elemen-elemen yang saling berhubungan membentuk satu kesatuan untuk melaksanakan suatu tujuan pokok dari sistem tersebut. Atau dapat diartikan pula sebagai suatu jaringan kerja dari *prosedur-prosedur* yang teratur, saling berhubungan dan bersama-sama melakukan kegiatan untuk menyelesaikan sasaran tertentu.

Menurut Jogiyanto (2010:2) berpendapat bahwa: *Sistem* adalah kumpulan elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Menurut Edhy Sutanta (2010:4), Sistem adalah kumpulan hal atau *elemen* yang saling bekerja sama atau yang dibutuhkan dengan cara-cara tertentu sehingga membentuk suatu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan.

Jadi sistem adalah kumpulan hal atau *elemen* yang saling bekerja sama atau yang dibutuhkan dengan cara-cara tertentu sehingga saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu

2.2 Pengertian Informasi

Menurut Kristanto (2010:7), Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi.

Menurut Edhy Sutanta (2010:8), Informasi adalah data yang diolah menjadi bentuk yang berguna dan menjadi berarti bagi penerimanya.”

Jadi informasi adalah kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.3 Sistem Informasi

Menurut Kristanto (2010:7),

“Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari *komponen-komponen* dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.”

Menurut Tata Sutabri (2010:36),

“*Sistem* informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat *manajerial* dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.”

Jadi sistem informasi adalah suatu sistem yang dibuat manusia yang terdiri dari komponen-komponen yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dan kegiatan strategi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

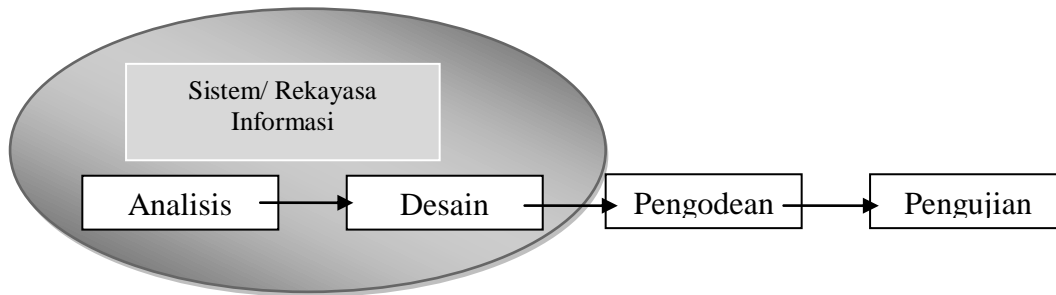
2.4 Metode Pengembangan Sistem

2.4.1 Waterfall

Menurut A.S. Rosa dan Shalahuddin.M(2011:26):

“*Waterfall* merupakan metodologi pengembangan sistem yang Menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut di mulai dari analisis, desain, pengodean, pengujian dan tahap pendukung. ”

Siklus pengembangan sistem menggunakan model *waterfall* pada gambar 2.1 :



Gambar 2.1 Paradigma *Waterfall*
Sumber : A.S, Rosa, Salahudin.M: (2011:27)

2.4.2. Tahapan – Tahapan Dalam Proses *Waterfall*

Menurut A.S. Rosa dan Shalahuddin. M (2011:27), Terdapat tahapan dalam proses *waterfall* yaitu:

1. *Analisis* Kebutuhan Perangkat Lunak, yaitu proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak seperti apa yang dibutuhkan user.
2. *Desain* Perangkat Lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean.
3. *Pengkodean* adalah desain harus ditranlasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap *desain*.
4. *Pengujian* adalah pengujian fokus pada perangkat lunak secara dari segi *logik* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

2.4.3. Kelemahan *Waterfall*

Menurut A.S. Rosa dan Shalahuddin.M(2011:28), Model air terjun dapat dilakukan sesuai alurnya karena sebab berikut:

1. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
2. Sangat sulit bagi pelanggan untuk mendefinisikan semua *spesifikasi* di awal alur pengembangan.
3. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan

2.4.4. Kelebihan *Waterfall*

Menurut A.S. Rosa dan Shalahuddin.M(2011:29):

“Hal positif dari model air terjun adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan (tidak ada tumpang tindih pelaksanaan tahap).”

2.5. Pengertian UML (*Unified Modelling Language*)

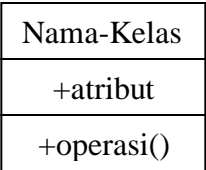



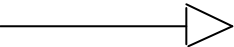
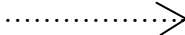
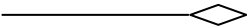
Menurut Nugroho, Bunafit (2005:17)

“UML (*Unified Modeling Language*) adalah proses penggambaran informasi-informasi dengan notasi-notasi baku yang telah disepakati sebelumnya”.

Diagram *UML* terdapat 9 diagram yaitu :

1. *Diagram Kelas*. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka, *kolaborasi-kolaborasi*, serta *relasi-relasi*. Diagram ini umum dijumpai pada pemodelan sistem *berorientasi objek*. Berikut adalah *simbol-simbol* yang ada pada diagram kelas dapat dilihat pada tabel 4.1 di bawah:

Table 2.1 Simbol Class Diagram

Simbol	Deskripsi
<p>Kelas</p> 	Menjelaskan tentang kelas pada struktur sistem yang akan dibuat
<p>Antarmuka/Interface</p>  <p>Nama interface</p>	Bagian dari <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi/asocition</p> 	Menjelaskan relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> pada sistem
<p>Asosiasi berarah/directed asocition</p> 	Menjelaskan tentang relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> pada sistem
<p>generalisasi</p> 	Menjelaskan tentang relasi antar kelas dengan makna <i>generalisasi-spesialisasi</i> (umum khusus) pada sistem.
<p>Kebergantungan/dependency</p> 	Menjelaskan tentang relasi antar kelas dengan makna kebergantungan antar kelas pada sistem
<p>Agregasi/aggregation</p> 	Menjelaskan tentang relasi antar kelas dengan makna semua bagian (<i>whole-part</i>) pada sistem.


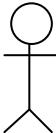
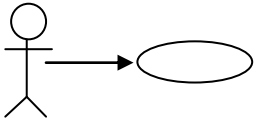
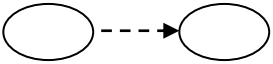
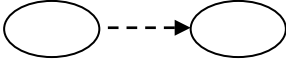
Sumber: Rosa A.S dan M.Shalahuddin (2013:205-206)

2. *Diagram Objek*. Bersifat statis. Diagram ini memperlihatkan *objek-objek* serta relasi-relasi antar objek. *Diagram objek* memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.

3. *Use-case diagram*. Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan *aktor-aktor* (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan.

Use-case diagram digambarkan dengan menggunakan *simbol-simbol* yang dapat dilihat pada tabel sebagai berikut :

Tabel 2.2 Simbol Diagram Use-Case

Simbol	Keterangan
	<i>UseCase</i> : menggambarkan bagian seseorang akan menggunakan/ memanfaatkan system.
	<i>Aktor</i> : seseorang/sesuatu yang berinteraksi dengan system yang kita kembangkan <i>Relasi</i> : sebagai penghubung antar actor-usecase, usecase-usecase dll.
	<i>Relasi Asosiasi</i> : relasi terjadi antar actor dengan usecase berupanya berupa garis lurus dengan kepala panah disalah satu ujung nya.
<p><<include>></p> 	<i>Include Relationship (Relasi Cakupan)</i> : memungkinkan suatu usecase untuk menggunakan fungsionalitas yang disediakan oleh usecase yang lainnya.
<p><<extend>></p> 	<i>Extend Relationship</i> : memungkinkan usecase memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh usecase yang lainnya.

Sumber : Munawar(2005 : 63)

4. *Sequence Diagram*. Bersifat dinamis. Diagram urutan adalah interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu. *Sequence Diagram* digambarkan dengan menggunakan simbol-simbol yang dapat dilihat pada tabel 4.3 sebagai berikut:

5. *Colaboration Diagram*. Bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. *Statechart Diagram*. Bersifat dinamis. Diagram state ini memperlihatkan *statestate* pada sistem, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamsi dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. *Activity Diagram*. Bersifat dinamis. Diagram aktivitas ini adalah tipe khusus dari digram states yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi.
8. *Component Diagram*. Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
9. *Deployment Diagram*. Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan saat *run time*. Diagram ini membuat simpul-simpul (*node*) beserta komponen-komponen yang ada di dalamnya.

1. *MySQL*

Menurut Bunafit Nugroho (2008:91)

“*MySQL (My Structured Query Language)* adalah sebuah program pembuat dan pengelola database atau yang sering disebut dengan DBMS (*DataBase Management System*, sifat dari DBMS ini adalah *Open Source*”.

MySQL adalah suatu sistem relational database yang menyimpan data pada tabel berbeda dan tidak meletakkannya pada satu tabel saja. Hal ini meningkatkan kecepatan dan *fleksibilitas*.

Tabel tersebut dihubungkan dengan suatu relasi yang didefinisikan sehingga dapat mengkombinasikan data dari beberapa tabel pada suatu saat. *MySQL* menggunakan standar (*Structure Query Language*), yaitu bahasa standar yang paling banyak digunakan untuk mengakses *database.MySQL* sebagai *database server* yang memiliki konsep *database modern* dan memiliki banyak sekali keistimewaan. Berikut ini adalah keunggulan dari *MySQL*:

1. *Portability*

MySQL dapat digunakan dengan stabil tanpa kendala, berarti pada berbagai sistem operasi diantaranya seperti *Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-UX*, dan masih banyak lagi.

2. *Open source*

MySQL didistribusikan secara *open source* (gratis) dibawah lesensi GPL, sehingga dapat digunakan secara Cuma-Cuma tanpa dipungut biaya.

3. *Multiuser*

MySQL dapat digunakan untuk menangani beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini akan memungkinkan sebuah *database server MySQL* dapat diakses *client* secara bersamaan dalam waktu yang bersamaan pula.

4. *Performance tuning*

Dalam hal performansi, *MySQL* memiliki kecepatan cukup menakjubkan dalam menangani *query-query* sederhana, serta mampu memproses lebih banyak SQL persatuan waktu.

5. *Column Types*

MySQL mendukung tipe kolom (tipe data) yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set*

serta enum. Bukan tidak mungkin tipe-tipe data tersebut akan terus dikembangkan oleh pengembang *MySQL*.

6. *Command dan Functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

7. *Security*

Sekuritas sistem merupakan hal yang sangat diperhatikan pada *MySQL*. Terbukti adanya beberapa lapisan sekuritas seperti level *subnetmask*, *namahost*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password terinkripsi*.

8. *Scalability dan Limits*

Dalam hal batasan kemampuan, *MySQL* terbukti mampu menagani *database* dalam skala cukup besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.

9. *Conectivity*

MySQL mampu melakukan koneksi dengan *client* menggunakan protocol *TCP/IP*, *Unix socket* (*Unix*), atau *Named Pipes* (*NT*).

10. *Localisation*

Localisation merupakan salah satu kemampuan *MySQL* dalam mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa. Dengan demikian, pengguna cukup menyesuaikan bahasa yang digunakannya

11. *Interface*

Sama halnya dengan *software-software database* lainnya, *MySQL* memiliki *interface* (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application programming Interface*).

12. *Client dan Tools*

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*. Hal ini akan semakin memudahkan pengguna dalam menggunakan *MySQL*.

13. Struktur Tabel

Struktur tabel *MySQL* cukup baik, serta cukup *fleksibel*, misalnya ketika menangani *ALTER TABLE* dibandingkan *database* lainnya semacam *PostgreSQL* ataupun *Oracle*.