

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

2.1.1 Pengertian Sistem

Pengertian Sistem sendiri merupakan deretan orang yang saling bekerja sama dengan ketentuan yang mempunyai anggaran yg sudah tersusun secara sistematis & terstruktur membangun kesatuan yang melaksanakan suatu tujuan tertentu [1].

Sistem juga adalah suatu jaringan kerja mekanisme-mekanisme yang saling berhubungan, berkumpul beserta untuk melakukan suatu aktivitas yang mempunyai tujuan tertentu [2].

Dari sumber literatur yang lain, Sistem merupakan sebuah tatanan (keterpaduan) yang tergabung atas sejumlah komponen fungsional (dengan fungsi & tugas khusus) yang saling berinteraksi dan secara bersama-sama bertujuan untuk memenuhi proses tertentu [3].

Dari beberapa pengertian tersebut dapat disimpulkan bahwa sistem adalah kumpulan elemen yang memiliki hubungan yang sama dan saling bekerja sama untuk membentuk suatu kesatuan dalam melaksanakan kegiatan yang memiliki tujuan tertentu.

2.1.2 Pengertian Informasi

Informasi adalah data yang telah diproses menjadi suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya. [4]

2.1.3 Pengertian Sistem Informasi

Sistem informasi yang sering kali dianggap sebagai sistem pemrosesan data adalah sistem yang dibentuk oleh manusia yang biasanya terdiri dari sekumpulan komponen yang saling terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi. [5]

2.2 Unified Modelling Language (UML) Diagram

2.2.1 Pengertian Unified Modelling Language

Unified Modelling Language adalah sebuah bahasa yang menjadi bahasa baku dalam suatu industri untuk memvisualisasikan, mendokumentasikan, dan merancang sistem piranti lunak. *UML* sendiri menawarkan sebuah standar baku untuk merancang sebuah model dalam perancangan sistem [6].

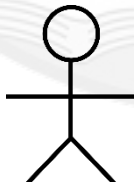

2.2.2 Diagram UML

UML menyediakan 4 jenis diagram yaitu :

1. Use Case Diagram

Use case diagram mendeskripsikan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan dalam *Use Case Diagram* adalah “apa” yang sistem perbuat, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara sistem dengan aktor yang bekerja. *Use case* sendiri adalah sebuah pekerjaan tertentu, contohnya *log in* ke sistem, membuat sebuah daftar inventaris, dan lain-lain. Seorang/sebuah aktor adalah sebuah entitas mesin atau manusia yang berinteraksi dengan sistem yang ada untuk melakukan pekerjaan-pekerjaan tertentu. Berikut ini merupakan elemen-elemen dari use case diagram:

Tabel 2.1 Elemen *Use Case Diagram*

Notasi	Nama Elemen	Keterangan
	<i>Actor</i>	Menggambarkan entitas yang ada bekerja dalam sistem
	<i>System Boundary</i>	Merepresentasikan batasan atau cakupan dalam sistem

Tabel 2.1 Elemen *Use Case Diagram* Lanjut

	<i>Use Case</i>	Merepresentasikan sistem berinteraksi dan juga menggambarkan kegiatan yang dilakukan oleh penggunaserta respons yang diberikan oleh sistem
	<i>Association</i>	Garis yang menghubungkan antara pengguna dan sistem saling berinteraksi
	<i>Include Relationship</i>	Hubungan pernyataan fungsi dari <i>use case</i> ke <i>use case</i> yang lain
	<i>Extend Relationship</i>	Merepresentasikan perluasan fungsi dari sebuah <i>use case</i>
	<i>Generalization Relationship</i>	Menghubungkan <i>use case</i> yang umum ke <i>use case</i> yang khusus

2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan suatu objek dan menjadi inti dari pengembangan dan desain

berorientasi objek. *Class* mendeskripsikan keadaan suatu sistem, dan juga menawarkan layanan untuk memanipulasi keadaan tersebut.

Class diagram menggambarkan struktur dan deskripsi dari *class*, *package* dan objek beserta hubungan antar satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Hubungan Antar *Class*

- a. Asosiasi, adalah hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain.
- b. Agregasi, adalah hubungan yang menyatakan bagian terdiri atas.
- c. Pewarisan, adalah hubungan hirarkis antar *class*. *Class* sendiri dapat diturunkan dari *class* lain dan dapat mewarisi semua atribut dan metoda dari *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
- d. Hubungan dinamis, adalah rangkaian pesan yang dikirimkan dari satu *class* kepada *class* yang lain.

3. *Activity Diagram*

Activity diagram menggambarkan alur aktivitas yang terjadi dalam sistem yang sedang dibuat, alur dari awal sistem, keputusan yang mungkin terjadi, dan bagaimana alur berakhir. *Activity diagram* juga dapat menggambarkan proses dalam bentuk paralel yang mungkin terjadi pada beberapa eksekusi.

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

2.3 *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) merupakan sekumpulan peralatan atau cara untuk menjelaskan dan mendetailkan objek-objek atau data-data yang dibentuk berdasarkan objek yang berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan notasi.

Beberapa kelebihan ERD :

1. Mudah untuk dioperasikan dalam hal mencari data field yang dibutuhkan.
2. Mudah dalam hal melakukan analisa terhadap data.
3. Kemampuan dalam hal menghilangkan redudansi data [7].

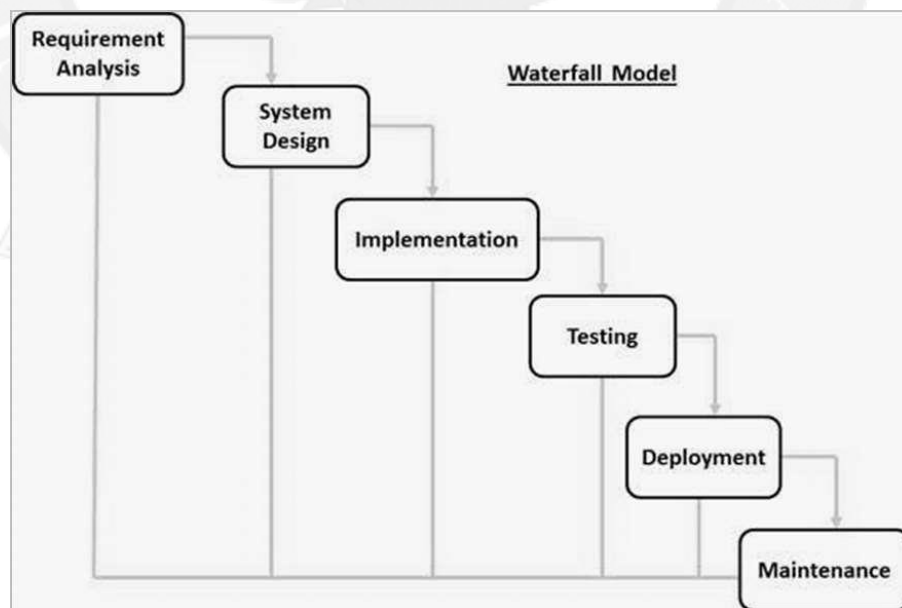
2.4 Software Development Life Cycle (SDLC)

Systems Development Life Cycle (SDLC) atau *Systems Life Cycle* pada rekayasa sistem dan perangkat lunak, merupakan proses perancangan & perubahan sistem serta metodologi dan menjadi contoh model yang digunakan untuk mengembangkan sistem-sistem tersebut [8].

SDLC sendiri terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

Model *Waterfall* merupakan salah satu contoh model pengembangan sistem perangkat yang bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan hingga tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan wajib dijalankan berurutan karena tahapan berikutnya tidak akan dilaksanakan sebelum tahapan yang dijalankan sebelumnya terselesaikan dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya.

Tahapan dari Model *Waterfall* ditunjukkan pada gambar di bawah ini :



Gambar 2.1 *Waterfall Model*

2.5 Sistem Informasi Berbasis Web

2.5.1 Pengertian Web

Website adalah perpaduan halaman yang menampilkan kumpulan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan menggunakan jaringan-jaringan halaman (*hyperlink*). [9]

2.5.2 Pengertian HTTP

Hypertext Transfer Protocol (HTTP) merupakan protokol pada *application-level* untuk sistem kolaboratif, terdistribusi, dan informasi hypermedia. HTTP mengartikan *poly* format & arti dari pertukaran pesan-pesan yang ada di internet dalam melakukan transaksi antara komponen Web, contohnya *client* dan *server*. HTTP mendefinisikan *syntax* berdasarkan tiap pesan yang ada dan bagaimana bagian per baris tiap pesan tersebut dapat diartikan. HTTP adalah protokol yang *stateless*, yaitu dimana *client* dan *server* tidak menyimpan informasi mengenai kondisinya ketika melakukan pertukaran *request* dan *response* [10].

2.5.3 Pengertian HTML

HyperText Markup Language (HTML) adalah Bahasa dalam pemrograman web yang digunakan untuk menggambarkan struktur halaman Web. HTML dapat digunakan untuk :

- a. Mempublikasi dokumen yang bersifat *online* dengan teks, tabel, foto, dan lain-lain.
- b. Mengambil informasi *online* melalui tautan *hypertext*, dengan sekali klik pada sebuah tombol.
- c. Membuat formulir untuk melakukan transaksi dengan untuk digunakan dalam mencari informasi, melakukan pemesanan, pembelian produk, dan lain-lain.
- d. Menyertakan klip video, klip suara, dan aplikasi lainnya secara langsung di dalam dokumen [11].

2.5.4 Pengertian PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat website dalam bentuk dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya dapat menampilkan konten statis, PHP dapat berinteraksi dengan database, folder dan file, sehingga PHP dapat menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, dan lain-lain. PHP sendiri merupakan bahasa scripting, tidak seperti HTML yang memiliki bahasa tag-based . PHP sendiri termasuk bahasa yang *cross-platform*, yang artinya PHP dapat berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac) [12].

2.5.5 Pengertian Database

Database adalah kumpulan file-file yang saling berinteraksi dan berkaitan satu sama lain, relasi tersebut jika ditunjukkan dapat diperlihatkan dengan kunci dari tiap-tiap file. Dalam satu database menunjukkan suatu kumpulan data yang dipakai dalam suatu ruang lingkup perusahaan, instansi. Pengolahan database sendiri merupakan suatu cara yang dilakukan terhadap file-file yang berada di suatu tempat yang dimana file tersebut dapat diurut, disusun, dan diambil sewaktu-waktu serta dapat ditampilkan dalam bentuk suatu laporan sehingga dapat mengolah file-file yang ada dan dapat memberikan informasi secara rapi.

2.5.6 Pengertian Sistem Informasi Berbasis Web

Sistem informasi berbasis web merupakan sekumpulan perangkat yang saling berhubungan untuk memproses suatu informasi dalam bentuk teks, gambar, suara yang ditampilkan dalam bentuk *hypertext* serta dapat diakses oleh perangkat lunak untuk membantu pembuatan kegiatan dalam organisasi untuk mencapai tujuan tertentu. Penerjemahan dokumen *hypertext* ke dalam bentuk yang dapat dipahami oleh manusia menggunakan *web browser* melalui *web client* yang tersimpan dalam *web server* [13].

2.5.7 Kelebihan Sistem Informasi Berbasis Web

Kelebihan Sistem Informasi Berbasis Web antara lain :

1. Dapat dijalankan dimanapun dan kapanpun tanpa harus melakukan proses penginstallan.

2. Tidak memerlukan lisensi apabila menggunakan web-based application, karena lisensi menjadi tanggung jawab web penyedia aplikasi.
3. Dapat dijalankan di semua sistem operasi.
4. Dapat diakses melalui berbagai perangkat.

2.5.8 Kekurangan Sistem Informasi Berbasis Web

Kekurangan Sistem Informasi Berbasis Web antara lain :

1. Membutuhkan koneksi internet yang stabil, hal ini bertujuan agar web yang ditampilkan dapat berjalan dengan baik.
2. Dibutuhkan sistem keamanan yang baik, karena aplikasi dijalankan secara terpusat.

2.5.9 *CodeIgniter*

CodeIgniter merupakan aplikasi *open source* berupa *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun aplikasi web dinamis dengan cepat dan mudah. *CodeIgniter* memiliki desain dan struktur file yang sederhana, didukung dengan dokumentasi yang lengkap sehingga *framework* ini lebih mudah dipelajari. *CodeIgniter* ini memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Artinya bahwa *CodeIgniter* 8 masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau dengan *syntax* umum didalam PHP, tidak harus menggunakan aturan penulisan kode di *CodeIgniter* [14].