



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Pengertian Data

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berupa suatu keadaan, gambar, suara, huruf, angka, matematika, Bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, objek, kejadian ataupun suatu konsep.

Data bisa merupakan jam kerja bagi karyawan perusahaan, data ini kemudian perlu di proses. Jika jam kerja setiap karyawan dikalikan dengan nilai per-jam, maka akan dihasilkan suatu nilai tertentu. Jika gambaran penghasilan setiap karyawan kemudian dijumlahkan, akan menghasilkan rekapitulasi gaji yang harus dibayar oleh perusahaan (Ganjarsayogo,2015).

2.2 Sistem Informasi

Sistem informasi merupakan suatu perkumpulan data yang terorganisasi beserta tata cara penggunaannya yang mencakup lebih jauh dari pada sekedar penyajian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tatacara penggunaannya. Keberhasilan suatu sistem informasi yang diukur berdasarkan

maksud pembuatannya tergantung pada tiga faktor utama yaitu: keserasian dan mutu data, pengorganisasian data, dan tata cara penggunaannya. Untuk memenuhi permintaan penggunaan tertentu, maka struktur dan cara kerja sistem informasi berbeda-beda bergantung pada macam keperluan atau permintaan yang harus dipenuhi. Suatu persamaan yang menonjol ialah suatu sistem informasi menggabungkan berbagai ragam data yang dikumpulkan dari berbagai sumber. Untuk dapat menggabungkan data yang berasal dari berbagai sumber suatu sistem alih rupa (*transformation*) data sehingga jadi tergabungkan (*compartible*). Berapapun ukurannya dan apapun ruang lingkungannya suatu sistem informasi perlu memiliki ketergabungan (*compatibility*) data yang disimpannya (Fatta,2009).

2.3 Aplikasi berbasis Web

Pada awalnya aplikasi *web* dibangun dengan hanya menggunakan bahasa yang disebut HTML (*HyperText Markup Language*). Pada perkembangan berikutnya, sejumlah skrip dan obyek yang dikembangkan untuk memperluas kemampuan HTML, seperti PHP dan ASP pada skrip dan *Applet* pada objek. Aplikasi *web* dapat dibagi menjadi dua jenis yaitu aplikasi *web* statis dan dinamis. *Web* statis dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang

terjadi. Kelemahan ini diatasi oleh model aplikasi *web* dinamis. Pada aplikasi *web* dinamis, perubahan informasi dalam halaman *web* dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi *web* dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggungjawab *webmaster*. Arsitektur aplikasi *web* meliputi klien, *web server*, *middleware* dan basis data. Klien berinteraksi dengan *web server*. Secara *internal*, *web server* berkomunikasi dengan *middleware* yang berkomunikasi dengan basis data. Contoh *middleware* adalah PHP dan ASP. Pada mekanisme aplikasi *web* dinamis, terjadi tambahan proses yaitu *server* menerjemahkan kode PHP menjadi HTML. Kode PHP yang diterjemahkan oleh mesin PHP yang akan diterima oleh klien (Kadir, 2009).

2.4 Website

Website adalah metode untuk menampilkan informasi-informasi di internet, baik itu merupakan teks, gambar, video, dan suara yang menghubungkan (*link*) dari dokumen satu dengan dokumen lainnya (*hypertext*) yang dapat di akses melalui *browser* (Yuhfizar, 2008).

Website adalah kumpulan halaman *web* yang saling terhubung dan file-filenya saling terkait. *Web* terdiri dari *page* atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas,

dengan halaman-halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam web (Agung, 2000).

Untuk dapat membuat sebuah *website* yang baik dan benar, kita perlu memahami beberapa bahasa pemrograman yaitu seperti:

2.4.1 Hyper Text Markup Language (HTML)

HTML yang biasa dikenal sebagai *web page*. HTML adalah sebuah intruksi-intruksi yang dibuat dan kemudian diterjemahkan oleh *browser* yang ada di komputer *client* sehingga isi formasinya dapat ditampilkan secara visual di komputer pengguna (Anamisa & Kustiyahningsih, 2010).

2.4.2 Personal HomePage (PHP)

PHP adalah bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat *server side* (Nugroho, 2006).

PHP adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan format HTML (Arief, 2011).

2.4.3 Hyper Text Transfer Protocol (HTTP)

HTTP adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan menyediakan dokumen yang diminta *browser* (Anamisa & Kustiyahningsih, 2010).

2.4.4 Javascript

Javascript adalah bahasa yang “*case sensitive*” artinya membedakan penamaan variabel atau fungsi yang menggunakan huruf besar dan huruf kecil (Anamisa & Kustiyahningsih, 2010).

2.4.5 Cascading Style Sheet (CSS)

CSS (*Cascading Style Sheet*) digunakan dalam kode HTML untuk menciptakan suatu kumpulan *style* yang dapat digunakan untuk memperluas kemampuan HTML, sebagai contoh kode HTML murni (tidak menggunakan CSS) tidak memungkinkan untuk mengatur ukuran font yang diterapkan pada setiap sel (Sutarman, 2009).

2.4.6 PhpMyAdmin

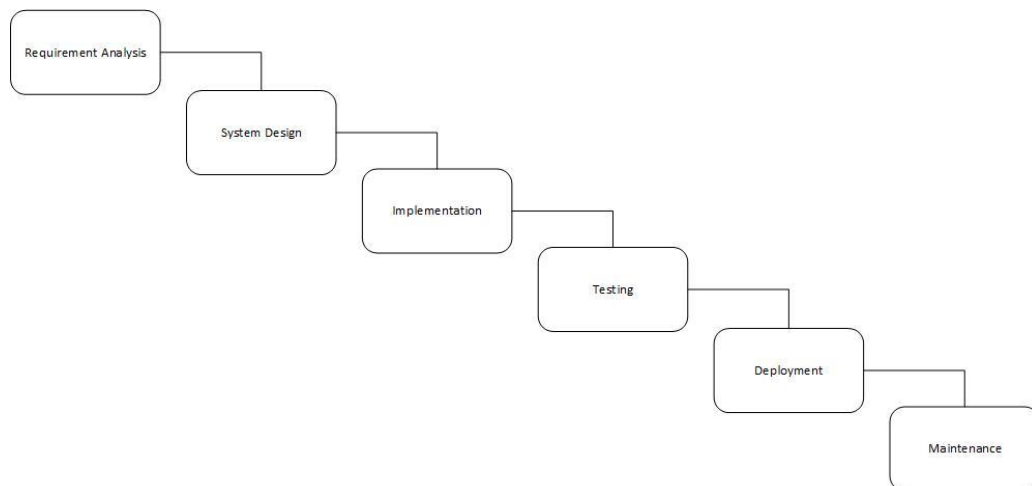
PhpMyAdmin adalah suatu aplikasi GUI (*Graphical User Interface*) yang digunakan untuk mengelola database MySQL (Arief, 2011).

2.5 Systems Development Life Cycle (SDLC)

Menurut Roger S. Pressman, 1992 . *SDLC Waterfall* atau metode yang memiliki nama lain “*Linear Sequential Model*” ini merupakan model yang paling banyak dipakai pada *Software Engineering* dan merupakan model yang muncul pertama kali pada tahun 1970. Model ini memberikan pendekatan-pendekatan sistematis dan berurutan bagi pengembangan piranti lunak. Pembangunan dengan metode *waterfall* memiliki tujuan yang berbeda-beda untuk setiap tahapan pembangunannya. Setelah tahapan pembangunan selesai, hasil pemngembangan ke tahap berikutnya dan tidak akan kembali ke tahapan yang sebelumnya.

Keuntungan dari menggunakan metode ini adalah memungkinkan adanya pengelompokkan kerja (*departementalization*) dan juga mengontrol manajerial. Setiap tahap pengembangan dapat diatur jadwalnya dan diberikan tenggat waktu, sehingga pembangunan akan lebih terstruktur dan selesai sesuai dengan rencana.

Sementara kerugian dari menggunakan metode ini adalah tidak memungkinkan adanya revisi. Setelah aplikasi sudah sampai pada tahap pengujian, sulit untuk kembali ke tahapan sebelumnya dan mengubah aplikasi, maka harus ada perencanaan yang matang pada saat pembangunan aplikasi dengan menggunakan metode ini.



Gambar 2.1 Contoh Model SDLC Waterfall

Berikut merupakan tahapan SDLC (*Software Development Life Cycle*) model *waterfall* :

1. Requirement Analysis (Analisis Kebutuhan)

Merupakan tahap analisis dan perencanaan dimana penulis mencari informasi mengenai permasalahan yang terkait dengan topik. Kebutuhan, kegunaan, dan batasan *software* juga harus didapatkan pada tahap ini. Informasi-informasi tersebut nantinya akan dianalisis untuk digunakan pada tahap selanjutnya.

2. System Design (Desain Sistem)

Merupakan tahap perancangan dimana penulis mencoba membuat solusi dari permasalahan yang diperoleh dari tahap analisis. Tahap ini membantu dalam memberikan spesifikasi kebutuhan

hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation* (Implementasi)

Merupakan tahap implementasi dimana penulis mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba. Tahap ini penulis melakukan pemrograman. Pada tahap ini pengembang juga melakukan pemeriksaan terhadap modul yang telah dibuat, dan menentukan apakah aplikasi yang dibangun sudah memenuhi fungsi yang diinginkan.

4. *Testing* (Pengujian)

Merupakan tahap dimana aplikasi di ujicoba dan memastikan bahwa aplikasi telah mampu menyelesaikan permasalahan yang telah didapat dari tahap *requirement analysis* dan mengecek apakah masih aplikasi masih terdapat kesalahan atau tidak.

5. *Deployment* (Penyebaran)

Merupakan tahap dimana aplikasi telah berhasil di ujicoba dan telah dipergunakan oleh banyak orang.

6. *Maintenance* (Pemeliharaan)

Merupakan tahap dimana pengguna telah menggunakan aplikasi dan melakukan perbaikan apabila terdapat masalah.

2.6 Sistem Peminjaman

Sistem dalam kamus besar Bahasa Indonesia berarti perangkat usur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas. Sistem merupakan sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan tujuan menerima masukan serta menghasilkan keluaran dalam proses transformasi yang teratur (Kadir,2003).

Peminjaman merupakan proses,cara, perbuatan meminjamn atau meminjamkan (Pusat Bahasa Departemen Pendidikan Nasional,2005).

Berdasarkan pengertian tersebut, tujuan sistem peminjaman adalah sekelompok komponen yang menyediakan fasilitas pengelolaan baik penyimpanan maupun pengelolaan data peminjaman. Diharapkan sistem dapat menghasilkan informasi secara cepat dan akurat.

2.7 *Entity Relationship Diagram (ERD)*

ERD (*Entity Relationship Diagram*) adalah merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analyst* dalam tahap analisis persyaratan proyek pengembangan sistem (Brady & Loonam, 2010).

Simbol-simbol dalam ERD :

1. Entitas (*Entity*)



Entitas adalah suatu objek yang dapat dibedakan dengan objek lainnya. Entitas berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama. Entitas memiliki bentuk persegi panjang.

2. Relasi/Hubungan Antar Entitas (*Relationship*)



Relasi adalah hubungan yang terjadi antara 1 entitas atau lebih yang tidak mempunyai fisik tetapi hanya sebagai konseptual. Dan berfungsi untuk mengetahui jenis hubungan yang ada antara 2 *file*. Relasi memiliki bentuk belah ketupat.

3. Atribut



Atribut adalah karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi

tersebut. Dan berfungsi untuk memperjelas atribut yang dimiliki oleh sebuah entitas. Atribut memiliki bentuk lingkaran lebih tepatnya elips.

4. Alur

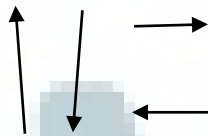
Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi. Alur memiliki bentuk garis.

2.8 Flowchart

Diagram *flowchart* merupakan diagram yang menggambarkan prosedur suatu program secara logis dan sistematis. *Flowchart* digunakan terutama dalam hal dokumentasi dan representasi suatu algoritma (J.Hartono,2005).

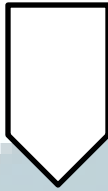
Simbol-simbol Flowchart:

- *Flow Direction Symbol*



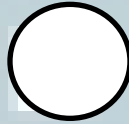
Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lainnya. Simbol ini disebut dengan *connecting line*.

▪ *Connector Symbol*



Simbol untuk keluar – masuk atau penyambung proses pada lembar / halaman yang berbeda.

▪ *Connector Symbol*

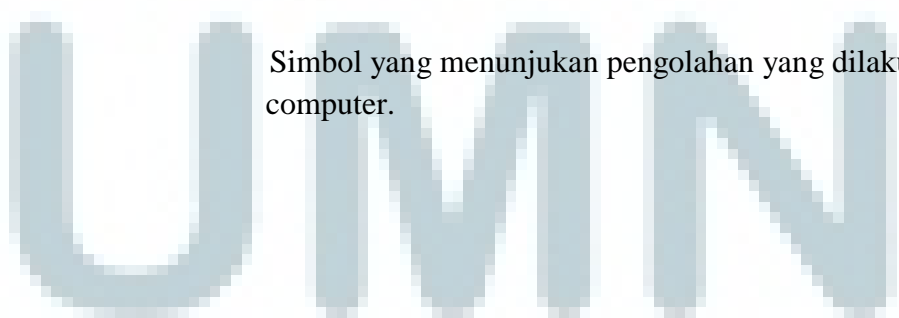


Simbol untuk keluar – masuk atau penyambung proses dalam lembar / halaman yang sama.

▪ *Processing Symbol*



Simbol yang menunjukkan pengolahan yang dilakukan oleh computer.

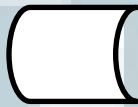


- *Symbol Decision*



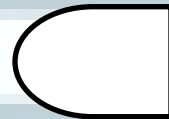
Simbol pemilihan proses berdasarkan kondisi yang ada.

- *Symbol disk and On-line Storage*



Symbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.

- *Symbol Display*



Symbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.

- *Symbol Input-Output*



Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.

- ***Symbol Manual Input***



Simbol untuk pemasukan data secara manual on-line keyboard.

- ***Symbol Manual Operation***



2.9 ***Data Flow Diagram (DFD)***

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikerjakan pada data tersebut (Kristanto, 2003).

DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem (Kristanto, 2003).

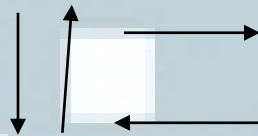
Simbol-simbol *data flow diagram*:

1. Kesatuan luar (*Eksternal Entity*)



Merupakan kesatuan diluar lingkuan system yang dapat berupa orang, organisasi atau system lain.

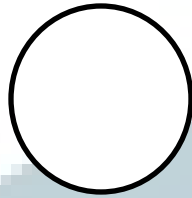
2. Arus Data



Merupakan arus dari data yang dapat berupa masukan untuk system atau hasil dari proses system.



3. Proses



Kegiatan atau kerja yang dilakukan orang, mesin atau computer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

4. Simpanan Data

Simpanan data dapat berupa suatu file, arsip catatan, buku, symbol catatan.

UMMN