

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

Menurut Gordon B. Davis dalam Zakiyudin (2012:5) “Sistem merupakan seperangkat unsur-unsur yang terdiri dari manusia, alat, konsep dan prosedur yang dihimpun menjadi satu untuk maksud dan tujuan bersama”. Sedangkan menurut Raymond McLeod Jr dalam Zakiyudin (2012:5) “Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan”.

Menurut Darmawan dan Kunkun Nur Fauzi (2013:2) Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut.

Dari pengertian sistem dan informasi diatas maka dapat kita simpulkan bahwa sistem informasi adalah seperangkat unsur-unsur yang berisi data yang telah diolah sehingga memiliki arti dengan maksud untuk mencapai tujuan yang telah ditentukan bersama.

B. Website

Menurut menurut Kustiyahningsih (2011:113) “*Web* adalah layanan yang didapat oleh pemakai komputer yang terhubung ke *internet*”.

Menurut Hidayat (2010:2) *website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk

satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

Menurut Hidayat (2010:2) jenis-jenis *web* berdasarkan sifat atau *stylenya* yaitu:

1. *Website* Dinamis, merupakan sebuah *website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan *database* Mysql atau MS SQL.
2. *Website* Statis, merupakan *website* yang kontennya sangat jarang berubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan database.

Berdasarkan pada fungsinya, *website* terbagi atas:

1. *Personal website*, *website* yang berisi informasi pribadi.
2. *Commercial website*, *website* yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
3. *Government website*, *website* yang dimiliki oleh instansi pemerintah, pendidikan, yang bertujuan memberikan pelayanan kepada pengguna.
4. *Non-profit Organization website*, dimiliki oleh organisasi yang bersifat *nonprofit* atau tidak bersifat bisnis.

C. Surat

Menurut Finoza (2009:4) “Surat adalah informasi tertulis yang dapat dipergunakan sebagai alat komunikasi tulis yang dibuat dengan persyaratan tertentu”. Persyaratan tertentu yang sekaligus menjadi ciri pembeda antara surat dan karangan tertulis lainnya adalah dalam hal:

1. Penggunaan kertas (sebagian surat menggunakan kertas khusus).

2. Penggunaan model atau bentuk (*style*).
3. Penggunaan kode dan notasi.
4. Pemakaian bahasa yang khas.
5. Pencantuman tanda tangan.

D. Unified Modelling Language (UML)

Menurut Nugroho (2010:6) “UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’”. Sedangkan menurut Herlawati dan Prabowo Pudjo Widodo (2011:10) “Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”.

Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram kelas (*Class Diagram*)

Bersifat statis, Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

2. Diagram paket (*Package Diagram*)

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. Diagram use-case (*Usecase 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 suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *sequence* (*Sequence Diagram*)

Bersifat dinamis. Diagram urutan adalah iterasiksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

5. Diagram komunikasi (*Communication Diagram*)

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

6. Diagram *statechart* (*Statechart Diagram*)

Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktivitas.

7. Diagram aktivitas (*Activity Diagram*)

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. Diagram komponen (*Component Diagram*)

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

9. Diagram *deployment* (*deployment diagram*)

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram, dan sebagainya.

E. **Entity Relationship Diagram (ERD)**

Menurut Al-Fatta (2009:27) “ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”. ERD merupakan model jaringan data yang menekankan pada struktur dan hubungan antar data. ERD juga memperlihatkan hubungan antar data *store* pada DFD.

Ada beberapa komponen yang terdapat dalam ERD, yaitu :

1. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data.
2. Atribut
 - a) Informasi yang diambil tentang sebuah entitas.
 - b) Nama atribut harus merupakan kata benda.
 - c) Kadang nama entitas diletakkan didepan nama atribut untuk ketelitian.
3. *Relationship*
 - a) Hubungan antar entitas.
 - b) Entitas pertama dalam *relationship* disebut entitas induk, entitas kedua disebut dengan entitas anak.

- c) *Relationship* harus memiliki nama yang berupa kata kerja
- d) *Relationship* berjalan 2 arah.

4. Kardinalitas

- a) Kardinalitas mengacu pada berapa kali *instance* dari suatu entitas.
- b) Satu *instance* dalam suatu entitas mengacu pada satu dan hanya satu *instance* pada entitas lainnya (1:1).
- c) Satu *instance* dalam suatu entitas mengacu ke satu atau lebih *instance* yang berelasi (1:N).
- d) Satu atau lebih *instance* dalam suatu entitas mengacu ke satu atau lebih *instance* yang berelasi (M:N).

F. *Logical Relationship Structure (LRS)*

Menurut Wulandari (2013:15) menerangkan bahwa:
Logical Record Structure dibentuk dengan nomor tipe record. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. LRS terdiri dari *link-link* diantara tipe *record*. *Link* ini menunjukkan arah dari satu tipe *record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link type record*.

Berikut tahapan transformasi ERD ke LRS:

1. Konversi ERD ke LRS, *Entity Relationship Diagram* harus diubah ke bentuk LRS (struktur *record* secara *logic*). Dari bentuk LRS inilah yang nantinya dapat ditransformasikan ke bentuk relasi tabel.
2. Konversi ERD ke LRS sebuah model sistem yang digambarkan dengan sebuah ERD akan mengikuti pola pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:
 - a. Setiap entitas diubah ke bentuk kotak dengan nama entitas, berada diluar kotak dan atribut berada didalam kotak.

- b. Sebuah *relationship* kadang disatukan, dalam sebuah kotak bersama entitas, kadang sebuah kotak bersama-sama dengan entitas, kadang disatukan dalam sebuah kotak tersendiri.
- c. Konversi LRS ke relasi tabel atau tabel adalah bentuk pernyataan data secara grafis dimensi, yang terdiri dari kolom dan baris. Relasi adalah bentuk visual dari sebuah *file*, dan tiap *tuple* dalam sebuah *field*, atau dalam bentuk lingkaran Diagram *Entity Relationship* dikenal dengan sebutan atribut. Konversi dari *logical structure*, dilakukan dengan cara:
 - 1) Nama *logical record structure* menjadi nama relasi.
 - 2) Tiap atribut menjadi sebuah kolom didalam relasi.

G. Web Server

Menurut Supardi (2010:2) “*Web server* merupakan perangkat lunak yang mengelola (mengatur) permintaan *user* dari *browser* dan hasilnya dikembalikan ke *browser*. Contoh *web server*, adalah IIS (*internet information services*) produk *microsoft corp*”.

H. Web Browser

Menurut Limantara (2009:1) “*Web Browser* adalah aplikasi perangkat lunak yang memungkinkan penggunaanya untuk berinteraksi dengan teks, *image*, *video*, *games* dan informasi lainnya yang berlokasi pada halaman *web* pada *World Wide Web* (WWW) atau *Local Area Network* (LAN).” Teks dan *image* pada halaman *web* dapat berisi *hyperlink* ke halaman *web* lain pada *website* yang sama maupun berbeda.

I. *Internet*

Menurut Hendartmoko (2009:9) mengemukakan bahwa “*Internet* merupakan sebuah sistem yang menghubungkan komputer satu dengan komputer lainnya dalam bentuk jaringan dan saling berinteraksi serta bertukar informasi”.

Sedangkan menurut Irawan (2011:102) mengemukakan bahwa “*Internet* merupakan kependekan dari “*Internetwork*”, yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan”.

Secara umum *internet* dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua informasi bisa didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan.

J. *Bahasa Pemrograman*

Bahasa pemrograman yang digunakan penulis pada pembuatan *website* pengajuan pembuatan surat di skripsi ini, yaitu :

1. PHP

Menurut Oktavian (2010:31) “PHP (*PHP Hypertext Prosesor*) adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang di gunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”.

Kode PHP mempunyai ciri-ciri khusus, yaitu:

- a. Hanya dapat dijalankan menggunakan *web server* misalnya: *Apache*.
- b. Kode PHP dapat diletakan dan dijalankan di *web server*.
- c. Kode PHP dapat digunakan untuk mengakses data bases, seperti: *MY SQL, PostgreSQL, Oracle*, dan lain-lain.
- d. Merupakan *software* yang bersifat *open source*.

- e. Gratis untuk di *download* dan digunakan.
- f. Memiliki sistem *multiplatform*, artinya dapat dijalankan menggunakan sistem operasi apapun, seperti *Linux*, *Unix*, *Windows*, dan lain-lain.

2. HTML

Menurut Oktavian (2010:13) “HTML (*Hypertext Markup Language*) adalah suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*).”

HTML mempunyai aturan dan struktur tertentu untuk menuliskan perintah-perintahnya yang biasa dinamakan dengan TAG HTML. Aturan tersebut diawali dengan lambang `<tag>` dan biasanya akan diakhiri dengan lambang `</tag>`.

3. Dreamweaver

Menurut Andi (2011:13) “*Dreamweaver* adalah sebuah HTML *editor profesional* untuk mendesain *web server visual* dan mengelola situs atau halaman *web*.” *Dreamweaver* merupakan *software* utama yang digunakan oleh *web designer* maupun *web programmer* dalam mengembangkan suatu situs *web*, karena *dreamweaver* mempunyai fasilitas yang mampu meningkatkan produktifitas dan efektifitas dalam desain *web*.

K. Model Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan yaitu Model *Waterfall*. Menurut Sukamto dan M. Shalahuddin (2013:28) menjelaskan bahwa “model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau

terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*). Berikut penjelasannya:

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

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 pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Tahap Pendukung

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Pada tahap ini proses akan dilakukan kembali sesuai urutan metode *waterfall*.

L. MYSQL

Menurut Anhar (2010:45) “MySQL (*My Structure Query Language*) adalah salah satu *DataBase Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lainnya.” MySQL berfungsi untuk mengolah *database* menggunakan bahasa *SQL*. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung dengan *database MySQL*.

M. Database

Menurut Oktavian (2010:62) “*Database* adalah sekumpulan data dan prosedur yang memiliki struktur sedemikian rupa sehingga mudah dalam menyimpan, mengatur dan menampilkan data.”

Beberapa komponen yang terdapat di *database* yaitu:

1. *Table* adalah sekumpulan data dengan struktur yang sedemikian rupa terbentuk dari *record* dan *field*. Sebuah tabel dapat terdiri dari beberapa *field* dan *record*.
2. *Record* adalah sekumpulan *field* yang membentuk suatu objek tertentu.
3. *Field* adalah atribut dari objek yang memiliki tipe data tertentu.

N. *Blackbox Testing*

Menurut Sukamto dan M. Salahuddin (2014:275) “*Blackbox testing* adalah perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Penguji dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus *login* maka kasus uji yang dibuat adalah:

- 1) Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- 2) Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.2 Penelitian Terkait

Menurut Rotikan dan Calvin Tamboto (2016:2) diharapkan dengan adanya sistem informasi ini, dapat meningkatkan pelayanan dari aparat pemerintah untuk masyarakatnya dalam rangka pengurusan surat-surat keterangan. Selain itu sistem ini juga dapat digunakan warga untuk mengajukan permohonan pembuatan surat, sehingga masyarakat hanya perlu datang sekali saja ke kantor Desa untuk mengambil surat yang diminta.

Menurut Yudhanto dkk (2017:1) Harapan menggunakan sistem informasi desa adalah agar kegiatan administrasi kependudukan serta pelayanan kepada masyarakat dapat dilakukan pada satu proses menjadi lebih cepat dan kevalidan data yang akurat. Tugas utama sistem ini adalah untuk membantu dalam mengelola data kependudukan, serta dilengkapi modul atau fitur untuk membuat laporan rekap data, membuat surat keterangan dan pengantar, serta membackup data kependudukan.