

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar Web

Dalam penulisan tugas akhir ini diperlukan teori yang relevan atau benar untuk mendukung kemudahan dalam mempelajari serta merancang *web* yang diharapkan dapat berjalan secara maksimal dan mudah dimengerti. Dengan demikian, program ini akan sangat membantu untuk mendapatkan informasi yang dibutuhkan, serta mudah diakses dengan cepat oleh pengguna.

##### A. *Website*

Menurut Yuhefizar dalam Agus & Safitri (2015:2) “*Website* adalah keseluruhan halaman-halaman *web* yang terdapat dari sebuah domain yang mengandung informasi”.

##### 1. *Web*

Menurut Arief dalam Kurniawati & Ghofar (2015:2) “*Web* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, animasi, video) didalamnya yang menggunakan Protokol HTTP (*hypertext transfer protocol*) dan untuk mengakses menggunakan perangkat lunak yang di sebut *browser*”.

##### 2. *Internet*

Menurut Oneto dan Sugiarto dalam Agus & Safitri (2015:2) “*internet* adalah jaringan komputer”. Ibarat jalan raya, internet dapat dilalui berbagai sarana transportasi, seperti bus, mobil dan motor yang memiliki kegunaan masing-masing.

##### 3. *Web Server*

Menurut Fathansyah dalam Agus & Safitri (2015:2) “*Server Web (Web Server)* merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan

akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”.

#### 4. *Web Browser*

*Web browser* menurut Winarno dan Utomo dalam Agus & Safitri (2015:2) “*web browser* adalah alat yang digunakan untuk melihat halaman *web*”.

### B. Bahasa Pemrograman

Menurut Yatini dalam Sudarsono dan Erniyanti (2017:2) “Bahasa pemrograman adalah teknik komunikasi standar untuk mengekspresikan instruksi-instruksi kepada komputer”.

#### 1. *Hyper Text Markup Language (HTML)*

Menurut Rerung (2018:18) mengatakan bahwa: HTML adalah singkatan dari *HyperText Markup Language*. Disebut *hypertext* karena di dalam HTML sebuah *text* biasa dapat berfungsi lain, kita dapat membuatnya menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan mengklik *text* tersebut. Kemampuan *text* inilah yang dinamakan *Hyper Text*, walaupun pada implementasinya nanti tidak hanya *text* yang dapat dijadikan *link*. Disebut *Makup Languange* karena bahasa HTML menggunakan tanda (*mark*), untuk menandai bagian-bagian *text*.

#### 2. *Cascading Style Sheets (CSS)*

Menurut Rerung (2018:133) “CSS adalah singkatan dari *Cascading Style Sheets*. CSS biasanya selalu dikaitkan dengan HTML, karena keduanya memang saling melengkapi dimana HTML ditujukan untuk membuat struktur, atau konten dari halaman *web*. Sedangkan CSS digunakan untuk tampilan dari halaman *web* tersebut”.

#### 3. *Hypertext Preprocessor (PHP)*

Menurut (Rerung, 2018:156) mengatakan bahwa: PHP adalah singkatan dari “PHP: *Hypertext Preprocessor*“, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs *web*, mulai dari halaman *web* yang sederhana sampai aplikasi kompleks yang membutuhkan koneksi ke *database*. PHP merupakan *script* yang menyatu dengan HTML dan berada pada *server (server side HTML embedded scripting)*.

#### 4. Javascript

Menurut Sidik dalam Agus & Safitri (2015:2) “Menjelaskan bahwa JavaScript adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja”.

#### C. *Framework*

Menurut Wardana dalam Kesuma, Meisella Kristina, & Fatima (2018:28) *Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi *website* kita harus mengikuti aturan dari *framework* tersebut. *Framework* yang dimaksud adalah kumpulan dari beberapa fungsi yang di buat oleh orang atau perusahaan fungsinya adalah memudahkan pengembang agar terutama mengurangi waktu pembuatan sebuah *website*.

##### 1. Codeigniter

Menurut Supono & Putratama (2016:109) “CodeIgniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP”.

##### 2. Bootstrap

Menurut Abdulloh dalam Kesuma, Meisella Kristina, & Fatima (2018:29) “Bootstrap merupakan salah satu *framework* CSS yang sangat populer dikalangan pecinta pemrograman *website*. Dengan menggunakan bootstrap, proses desain *website* tidak dibuat dari nol, sehingga proses desain *website* lebih cepat dan mudah”.

##### 3. JQuery

Menurut Abdulloh dalam Kesuma, Meisella Kristina, & Fatima (2018:29) “jQuery merupakan salah satu javascript *library*, yaitu kumpulan fungsi javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode javascript”.

#### 4. DataTables

Menurut Purbadian dalam Kesuma, Meisella Kristina, & Fatima (2018:29) menyimpulkan bahwa “DataTables adalah sebuah plugin jquery yang sangat *power full*. Di samping mudah untuk mengimplementasikannya, *plugin* jquery datatables sudah terintegrasi dengan *search form*, *pagination*, serta jumlah data yang akan ditampilkan”.

#### D. Basis Data

Menurut Winarno dan Utomo dalam Agus & Safitri (2015:2) “*Database* atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling berhubungan satu sama lain, dengan menggunakan *field*/kolom pada tiap tabel yang ada”.

##### 1. Mysql

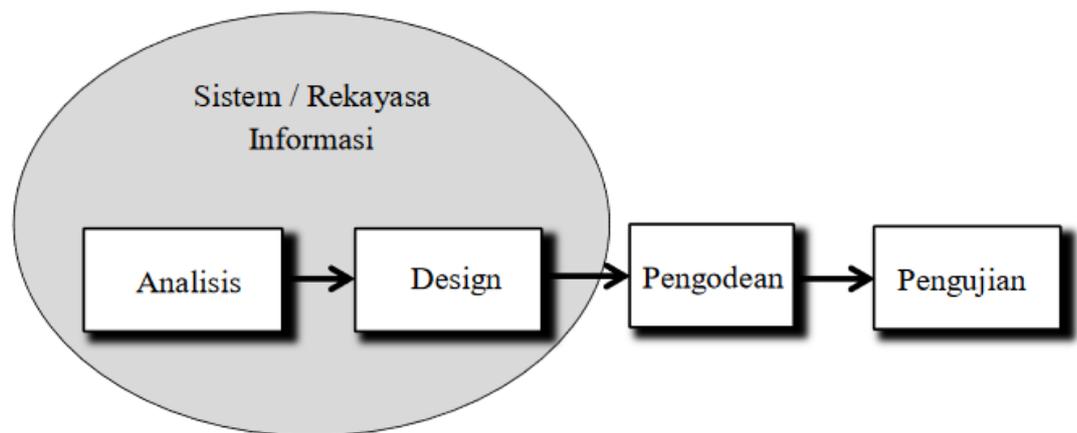
Menurut Padang dalam Kesuma, Meisella Kristina, & Fatima (2018:29) “MySQL merupakan komponen yang digunakan untuk mengakses atau berkomunikasi dengan database MySQL melalui PHP”. Beberapa manfaat penggunaan MySQL yaitu: Menggunakan *Interface*/metode *object oriented*, menggunakan penggunaan *prepared statement*, mendukung penggunaan *multiple statement*, mendukung penggunaan transaksi, peningkatan terhadap kemampuan pencarian kesalahan program, mendukung pengembangan dengan *server*.

##### 2. Phpmyadmin

Menurut Abdulloh dalam Kesuma, Meisella Kristina, & Fatima (2018:29) “Phpmyadmin merupakan aplikasi berbasis *web* yang digunakan untuk membuat *database* MySQL sebagai tempat untuk menyimpan data-data *website*”.

#### E. Model Pengembangan Perangkat Lunak

Menurut Sukamto & M. Shalahuddin (2014:28) “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)”. Berikut adalah gambar model air terjun:



Sumber : Sukamto & M. Shalahuddin (2014)

**Gambar II.1**

**Ilustrasi Model *Waterfall***

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 di dokumentasikan.

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 pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

### 3. Pembuatan Kode Program

Desain harus di translasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yangtelah dibuat pada tahap desain.

### 4. Pengujian

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

### 5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## 2.2. Teori Pendukung

Dalam penulisan tugas akhir ini penulis juga menambahkan teori pendukung yang berhubungan dengan konsep dasar pembuatan *web* dan diperlukan dalam sistem informasi perpustakaan berbasis *web* yaitu:

### A. Struktur Navigasi

Menurut Prihatna dalam Rohman, Ermawati, Farlina, & Syabaniah (2018:87) “Struktur Navigasi adalah susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan *link* atau navigasi halaman pada suatu situs *web*”.

Menurut Binanto dalam Eka & Arviana (2018:179) Ada empat struktur navigasi dasar yaitu:

1. Linear

Pengguna akan melakukan navigasi secara berurutan dari frame atau byte informasi yang satu ke yang lainnya.

2. Hierarkis

Struktur dasar ini disebut juga struktur “linear dengan percabangan” karena pengguna melakukan navigasi disepanjang cabang pohon struktur yang terbentuk oleh logika isi.

3. Nonlinear

Pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terkait dengan jalur yang sudah ditentukan sebelumnya.

4. Komposit

Pengguna akan melakukan navigasi dengan bebas (secara nonlinear), tetapi terkadang dibatasi presentasi linear film atau informasi penting dan atau pada data yang paling terorganisasi secara logis pada suatu hierarki.

### **B. Entity Relationship Diagram (ERD)**

Menurut Sukanto & M. Shalahuddin (2014:50) “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan Entity Relationship Diagram (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODBMS maka perancangan ERD tidak perlu dilakukan”.

## 1. Komponen ERD

Berikut ini komponen-komponen ERD terdiri dari:

### a) Entitas (*Entity*)

Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

### b) Atribut (*Attribute*)

*Field* atau kolom data yang butuh disimpan dalam suatu entitas.

### c) Atribut kunci primer

*Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

### d) Atribut multi nilai atau *multivalued*

*Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

### e) Relasi

Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

### f) Asosiasi (*association*)

Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B.

## 2. Derajat Relasi

Definisi derajat *relationship* menurut Al-Bahra dalam Mulia (2016:35)“*Relationship Degree* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.”

Berikut ini derajat *relationship* yang digunakan ERD:

### a) *Unary Relationship*

*Unary Relationship* adalah model relasi yang terjadi di antara *entity* yang berasal dari *entity set* yang sama. Sering juga disebut sebagai *recursive relationship* atau *relective relationship*.

### b) *Binary Relationship*

*Binary Relationship* adalah model relasi antar instansi-instansi (*instances*) dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

### c) *Tenary Relationship*

*Tenary Relationship* merupakan *relationship* antara instansi-instansi dari tiga tipe entitas secara sepihak. Masing-masing entitas mungkin berpartisipasi satu atau banyak dalam suatu *relationship ternary*. Perlu dicatat bahwa *relationship ternary* tidak sama dengan tiga *relationship binary*

## 3. *Logical Record Structure* (LRS)

Menurut Tabrani dalam Kesuma, Meisella Kristina, & Fatima (2018:29) “LRS adalah *Logical Record Structure* dibentuk dengan nomor dari tipe *record*”. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan E-R diagram adalah nama tipe *record* berada diluar kotak *field* tipe *record* ditempatkan.

### C. Implementasi dan Pengujian Web

Pada bagian implementasi dan pengujian web penulis menggunakan metode *blackbox testing* untuk pengujian aplikasi yang akan berfokus pada fungsi dari aplikasi yang telah dibuat.

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

Menurut Rachmaniah (2018:50) “Pada *Blackbox testing* rencana uji dikembangkan langsung spesifikasi program, yaitu setiap item pada program spesifikasi menjadi sumber uji, dan beberapa kasus uji dikembangkan dari sumber uji tersebut”.

