

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar Web

##### A. Website

Menurut Bekti (2015:35) menyimpulkan bahwa: Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, 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.

##### 1. Web server

Menurut Fathansyah (2018:466) menyatakan bahwa “*Server Web (Web Server)* merujuk pada perangkat keras 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”.

##### 2. Web Browser

Menurut Hidayatullah (2017:4) menyatakan bahwa “*Web Browser* adalah menguraikan kode HTML, menerjemahkannya serta menuliskan *teks* yang terkandung didalam halaman *web* ke layar”.

##### 3. Internet

Menurut Sugeng (2015:32) menyatakan bahwa “*internet* adalah definisi resmi dari *internet* secara *global* dan menggunakan TCP/IP sebagai protokol pertukaran paket (*packet switcing communication protocol*). Prinsip IP (*Internet Protocol*) mengubah jaringan yang tidak sejenis menjadi jaringan yang *homogen*, sebagai hasilnya adalah *internet*”

#### 4. *Adobe Dreamweaver CS6*

Menurut Bekti (2015:2) menyatakan bahwa :

Adobe Dreamweaver CS6 adalah aplikasi desain dan pengembangan web yang menyediakan editor WYSIWYG visual (bahasa sehari-hari yang disebut sebagai *Design view*) dan kode editor dengan fitur standar seperti *syntax highlighting*, *code completion* dan *code collapsing* untuk menghasilkan petunjuk kode untuk membantu pengguna dalam menulis kode.

Tampilan awal yang terdapat pada Adobe Dreamweaver CS6 :

- a. Bagian kiri, yaitu *Open a Recent Item* merupakan alternatif untuk membuka *file*, atau *recent file*. Untuk membuka *file*, pilih *Open* pada *Starter Page*.
- b. Bagian tengah, yaitu *Create New* digunakan untuk membuat halaman *web* baru. Terdapat beberapa macam tipe *file* untuk format halaman *web* seperti HTML, *Clodfusion* dan PHP (*Hypertext Preprocessor*).
- c. Bagian kanan, yaitu *Top Features* digunakan untuk mempelajari lebih jauh tentang penggunaan *Dreamweaver* seperti situs yang menyediakan *Tutorial CSS* dalam bentuk *video*.

#### B. Bahasa Pemrograman

##### 1. PHP

Menurut Sidik (2014:4) menyatakan bahwa “PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dikenal juga sebagai bahasa pemrograman *server side*”.

##### 2. *JQuery*

Menurut Bekti (2015:59) menyatakan bahwa “*JQuery* merupakan salah satu *Library* yang membuat program *web* disisi klien, tidak terlihat sebagai program

*javascript* biasa, yang harus secara *eksplisit* disisipkan pada dokumen web”.

### 3. *JavaScript*

Menurut Subagia (2018:14) menyatakan bahwa “*JavaScript* merupakan bahasa pemrograman web yang bersifat *Client Side Programming*, artinya pemrosesan dilakukan di sisi klien”.

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

Menurut Bekti (2015:47) menyatakan bahwa “*CSS (Cascading Style Sheets)* merupakan salah satu bahasa pemrograman *web* yang digunakan untuk mempercantik halaman web dan mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam”.

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

Menurut Darmawan (2014:13) menyatakan bahwa “*Hyper Text Markup Language (HTML)* yang memungkinkan terjadinya *koneksi (link)* dokumen yang satu dengan yang lain atau bagian dari dokumen yang satu dengan bagian yang lainnya, baik dalam bentuk *teks, visual* dan lain-lainnya”.

## C. Basis Data

Menurut Fathansyah (2018:3) mendefinisikan bahwa “Basis data (*database*) merupakan kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

### 1. *MySQL*

Menurut Mandar (2017:34) menyatakan bahwa “ *MySQL* merupakan sebuah *server database SQL* atau *DBMS* yang *multiuser, multithread* yang bersifat *Open Source* di bawah lisensi *GNU General Public Licenci (GPL)* ”. Karena

sifatnya yang *Open Source* maka *MYSQL* dapat dipergunakan dan didistribusikan untuk kepentingan *individu* maupun *corporate* tanpa memerlukan *lisensi* dari pembuatnya.

## 2. *PhpMyadmin*

Menurut Subagia (2018:69) menyatakan bahwa “*PhpMyadmin* mendukung berbagai fitur administrasi *MySQL* termasuk manipulasi *database, table, index* dan juga dapat *export* data ke dalam berbagai format data”.

*PhpMyadmin* merupakan *front-end MySQL* berbasis *web*. *PhpMyadmin* Dibuat dengan menggunakan *PHP*. *PhpMyadmin* juga tersedia dalam lebih dari 50 bahasa.

## 3. XAMPP

Menurut Ardhana (2017:1) menyatakan bahwa “XAMPP adalah perangkat lunak yang bebas digunakan”. XAMPP berfungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari *Apache HTTP server, MySQL database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.

## D. Model Pengembangan Perangkat lunak

Menurut Rosa A S dan M. Shalahuddin (2016:28) Model pengembangan *waterfall* (air terjun) terbagi menjadi lima tahapan, yaitu:

### 1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasi 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 pengkodean. 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 dari 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. 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.

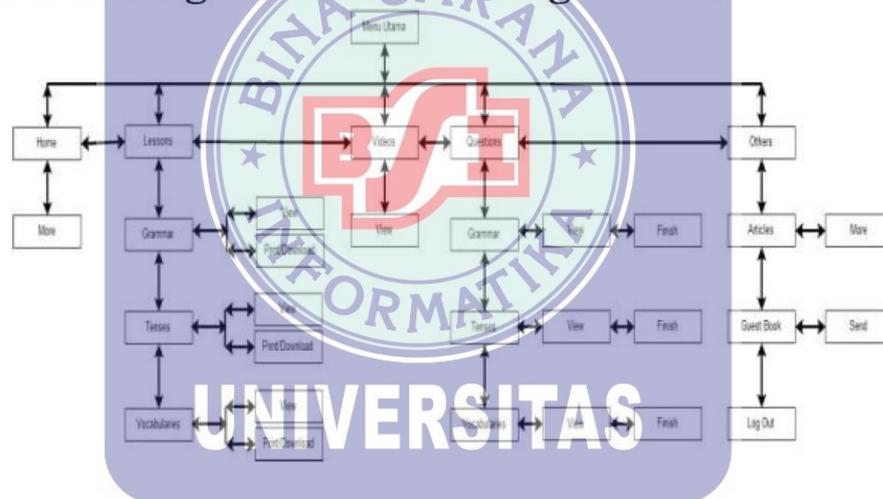
## 2.2. Teori Pendukung

### A. Struktur Navigasi

Menurut Rachmanto (2017:142) menjelaskan bahwa “ Struktur Navigasi atau menu navigasi merupakan bagian dari elemen *website* yang berisikan link-link penghubung utama yang akan mengarah pada halaman tertentu pada sebuah *website*”. Menentukan struktur navigasi merupakan halaman yang sebaiknya dilakukan sebelum membuat suatu *website*.

Contoh Struktur Navigasi :

### Rancangan Struktur Navigasi User



Sumber : (Amarulloh, Adiwisastro, & Mulyani, 2016)

**Gambar II.1. Stuktur Navigasi Sistem Informasi Edukasi Bahasa Inggris Berbasis Web**

### B. Enterprise Relationship Diagram

Menurut Fathansyah (2018:81) mendefinisikan bahwa “Model *Entity Relationship* yang berisi komponen-komponen himpunan Entitas dan himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasi seluruh fakta dari ‘dunia nyata’ yang kita tinjau, dapat digambarkan dengan lebih

sistematis dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R).”

1. Ada tiga komponen pembentuk Model *Entity-Relationship*, yaitu:

a. Entitas

“Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.” Sebuah kursi yang kita duduki, seseorang yang menjadi pegawai di sebuah perusahaan dan sebuah mobil yang melintas di depan kita adalah entitas.

b. Atribut

“Setiap Entitas pasti memiliki atribut yang mendeskripsikan karakteristik (properti) dari entitas tersebut. Sebagaimana telah disebutkan sebelumnya penentuan atribut-atribut yang relevan bagi sebuah Entitas merupakan hal penting lainnya dalam pembentukan model data”.

c. Relasi

“Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda”.

2. Kardinalitas Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas Relasi yang terjadi di antara dua himpunan entitas (misalnya A dan B) dapat berupa:

a. Satu ke Satu (*One to One*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

b. Satu ke Banyak (*One to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

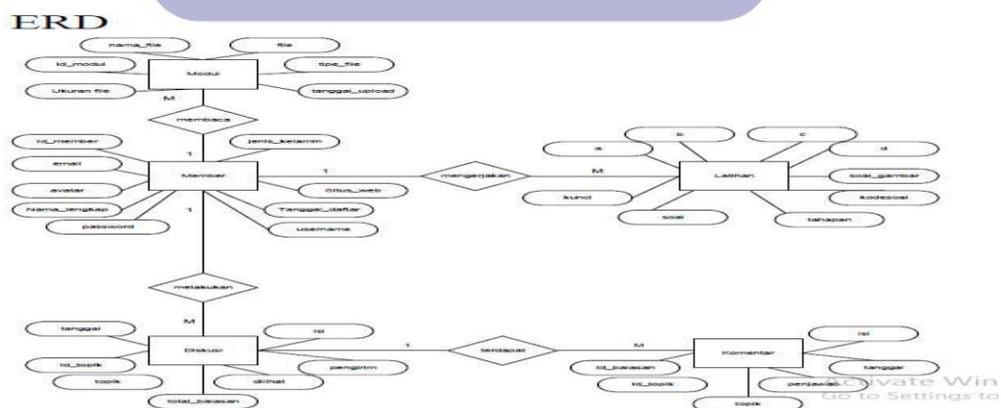
c. Banyak ke Satu (*Many to One*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B. Tetapi tidak sebaliknya, di mana setiap entitas himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

d. Banyak ke Banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, di mana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

Contoh ERD :



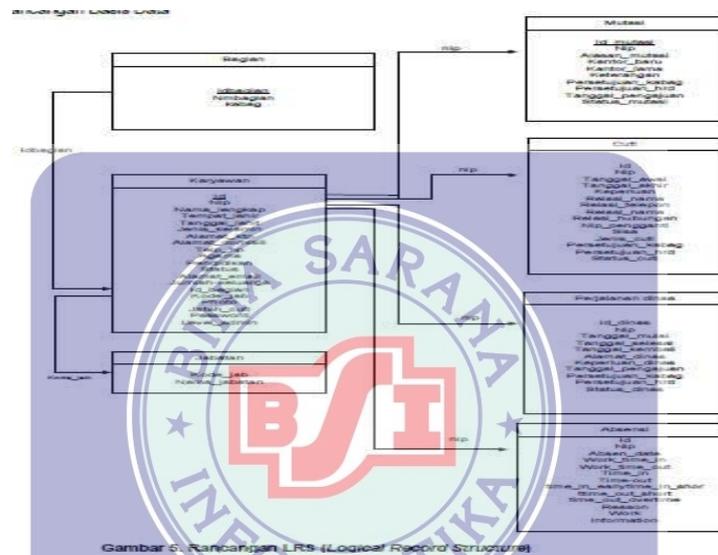
Sumber: (Zamaludin, Yusnaeni, Amelia, Pengembangan, & Lunak, 2016)

**Gambar II.2. ERD (*Entity Relationship Diagram*) Perancangan Pembelajaran Jarak Jauh (E-learning) Bahasa Jerman Berbasis Web**

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

Menurut Tabrani (2014:35) dalam jurnal (Hidayat, Marlina, & Utami, 2017) “*Logical Record Structure* terdiri dari *link-link* diantara tipe *record*, link ini menunjukkan arahan dari satu tipe *record* lainnya. Banyak link dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua link tipe *record*”.

Contoh LRS :



Sumber : (Sigalingging, Sriyadi, & Budiarti, 2016)

**Gambar II.3. LRS (*Logical Record Structure*) Sistem Informasi E-Kepegawaian Pada PT. Mannapratama Santosa Jakarta**

### C. Implementasi dan Pengujian Web

Sebuah perangkat lunak perlu dijaga kualitasnya bahwa kualitas bergantung kepada kepuasan pelanggan (*customer*). Sering perangkat lunak mengandung kesalahan (*error*) pada proses-proses tertentu pada saat perangkat lunak sudah berada di tangan *user*. Kesalahan-kesalahan (*error*) pada perangkat lunak ini sering disebut dengan “*bug*”. Untuk menghindari banyaknya *bug* maka diperlukan adanya pengujian perangkat lunak sebelum perangkat lunak diberikan ke pelanggan atau selama perangkat lunak masih terus dikembangkan.

Setelah pengujian sistem selesai dilakukan maka dapat dilakukan pengujian penerimaan perangkat lunak oleh pelanggan (*customer*) atau *user* (pemakai perangkat lunak). Pengujian penerimaan digunakan untuk mengetahui kepuasan pelanggan atau *user* terhadap perangkat lunak yang sudah dibuat. Jika pelanggan sudah puas dengan perangkat lunak, maka perangkat lunak dapat di serahkan kepada pelanggan (*customer*).

Menurut Rosa A S dan M. Shalahuddin (2016:275) menyimpulkan bahwa “Black-Box Testing (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi 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”.

Pengujian unit fokus pada usaha verifikasi pada unit yang terkecil pada desain perangkat lunak (komponen atau modul perangkat lunak). Setiap unit perangkat lunak diuji agar dapat diperiksa apakah aliran masukan (*input*) dan keluaran (*output*) dari unit sudah sesuai dengan yang diinginkan. Pengujian unit biasanya dilakukan saat kode program dibuat. Karena dalam sebuah perangkat lunak banyak memiliki Unit-unit kecil maka untuk menguji unit-unit kecil ini biasanya dibuat program kecil (*main program*) untuk menguji unit-unit perangkat lunak.

Unit disini secara fisik dapat berupa prosedur atau fungsi, sekumpulan prosedur atau fungsi yang ada dalam satu berkas (*file*) jika dalam pemrograman terstruktur, atau kelas, bisa juga kumpulan kelas dalam satu *package* dalam pemrograman berorientasi objek. setiap unit diuji menggunakan sebuah program pengujian yang khusus dibuat untuk menguji sebuah unit menggunakan kumpulan kasus uji yang didefinisikan.

Contoh Implementasi dan Pengujian Web :

**Tabel 1 Pengujian Form Login**

Skenario Pengujian	Test Case	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
Mengosongkan semua isian data <i>login</i> admin, lalu klik tombol ' <i>Login</i> '	Username: (kosong) Password: (kosong)	Sistem akan menolak dan muncul pesan "Anda belum mengisi Username"	Sesuai harapan	Valid
Mengisi data isian pada username, dan mengosongkan data password, lalu klik Login	Username: admin Password: (kosong)	Sistem akan menolak dan muncul pesan "Anda belum mengisi Password"	Sesuai harapan	Valid
Mengisi data isian pada password, dan mengosongkan data username, lalu klik Login	Username: (kosong) Password: (admin)	Sistem akan menolak dan muncul pesan "Anda belum mengisi Username"	Sesuai harapan	Valid

Sumber : (Hidayat, Marlina, & Utami, 2017)

**Gambar II.4. Contoh Pengujian Unit Perancangan Sistem Informasi Penjualan Barang *Handmade* Berbasis *Website* Dengan *Waterfall***

**UNIVERSITAS**