

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Pembuatan tugas akhir ini tidak lepas dari teori-teori yang mendukung kemudahan dalam mempelajari dan merancang program aplikasi yang diharapkan secara maksimal. Pada bab ini penulis mencoba mengemukakan beberapa bagian yang tertentu yang berkaitan serta mendukung dalam pelaksanaan pembuatan *website*.

A. Website

Menurut (Iqbal, 2019:4) “*Website* merupakan halaman yang menampilkan informasi data teks, gambar, suara, video, atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis”.

1. Internet

Menurut (Iskandar, 2018:233) “internet merupakan singkatan dari interconnection-networking yang merupakan sistem jaringan yang menghubungkan tiap-tiap computer secara global global *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP) sebagai protocol pertukaran paket (*packet switching communication protocol*) untuk melayani miliaran pengguna diseluruh dunia”.

2. Web Server

Menurut (Sandi, 2014:17) “aplikasi yang menajdi tulang belakang dari *World Wide Web* (WWW). *Web server* menunggu permintaan dari client yang menggunakan *web browser* seperti *Netscape Navigator*, *Internet Explore*, *Mozilla Firefox* dan Aplikasi web browser lainnya”.

3. *Web Browser*

Menurut (Abdulloh, 2018) menjelaskan bahwa “*Web browser* adalah untuk menampilkan dan mengetes hasil program”.

B. Bahasa Pemograman

1. PHP (*Personal Home Page*)

Menurut (Sidik, 2017:4) “PHP merupakan secara umum dikenal sebagai bahasa pemograman script-script yang membuat 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 pemograman server side”.

2. HTML (*Heypertext markup Language*)

Menurut (Abdulloh, 2016:2) “HTML singkatan dari *Heypertext markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur *website*”.

3. CSS (*Cascading Style Sheet*)

Menurut (Abdulloh, 2018:45) “CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai Property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan”.

4. *JavaScript*

Menurut (Abdulloh, 2018:193) “*Javascript* merupakan bahasa pemograman *web* yang pemrosesannya dilakukan disisi client”.

5. *Bootstrap*

Menurut (Abdulloh, 2018:261) “*Bootstrap* merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada”.

C. Basis Data

Menurut (Lubis, 2016:2) “Basis data merupakan gabungan file data yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat indenpenden”.

1. *SQL*

Menurut (Abdulloh, 2018:104) “*SQL* merupakan singkata dari *Structured Query Language* yaitu bahasa yang digunakan untuk mengakses dan manipulasi database”.

2. *Xampp*

Menurut (Andi, 2016:186) “*Xampp* adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla*, dan lain-lain. *Xampp* berfungsi untuk memudahkan instalasi lingkungan *PHP* , dimana biasanya lingkungan pengembangan *web* memerlukan *PHP*, *Apche*, *MySQL*, dan *PhpMyAdmin* serta *software-software* yang terkait dengan pengembangan *web*”.

3. *PhpMyAdmin*

Menurut (Andi, 2016:186) “*PhpMyAdmin* adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen *MySQL*”.

D. Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan metode *waterfall* (Shalahuddin, M dan Sukamto, 2016:28) “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensi linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*).” Model air terjun menyidiakan pendekatan alur hidup perangkat lunak secara sekuensial atau teurut dimulai dari analis, desain, pengodean dan tahap pendukung (*support*).

1. Analisa Kebutuhan Perangkat Lunak

Dalam menganalisa perangkat lunak penulis menjelaskan teori tentang perangkat lunak yang dibutuhkan dalam pembuatan *website* ini. Kebutuhan untuk *admin* membutuhkan beberapa *software* yang dibutuhkan seperti *sublime text* sebagai editor yang berfungsi membuat halaman *web*. *XAMPP* sebagai *web server* yang berfungsi untuk menjalankan hasil program, dan *PHP* (*hypertext preprocessor*) yang berfungsi untuk merancang dalam pembuatan *website*. Proses pengumpulan kebutuhan dilakukan secara intensif untuk menentukan kebutuhan perangkat lunak agar dapat dipahami seperti apa yang dibutuhkan oleh user.

2. Desain

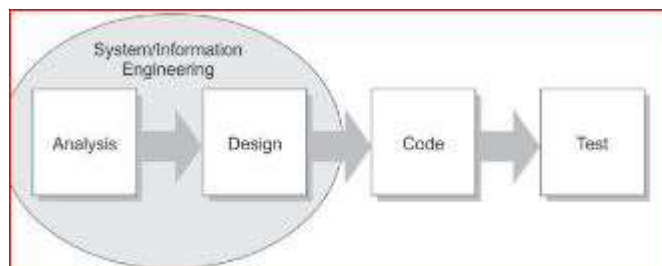
Dalam tahap ini perangkat lunak penulis gunakan ini meliputi, struktur navigasi sebagai petunjuk alur berjalan *web*, arsitektur, perangkat lunak, representasi antar muka, ERD (*Entity Relationship Diagram*) untuk merangkai data para anggota dan saling berhubungan berdasarkan objek.

3. Implementasi dan Penguji Unit

Tahap ini adalah untuk melakukan pengujian unit *logical* dan fungsional serta memastikan bahwa semua bagian *website* telah diuji, hal ini digunakan menggunakan *blackbox testing* agar tidak terjadi kesalahan (*error*) pada *website* dan menghasilkan keluaran yang diinginkan.

4. 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.



Sumber: (Shalahuddin, M dan Sukamto, 2016)

Gambar II.1
Metode *Waterfall*

2.2. Tools Program

Untuk lebih memahami isi dari tugas akhir ini, maka dibutuhkan beberapa pengetahuan mengenai definisi serta uraian yang berkaitan dengan teori pendukung sebagai berikut :

A. Struktur Navigasi

Menurut Prihatna dalam (Rohman, Ermawati, Farlina, dan Syabaniah, 2018) menjelaskan bahwa “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 dan Arviana, 2018) menjelaskan bahawa “ Ada 4 struktur dasar yang digunakan ,yaitu:

1. *Linear*

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

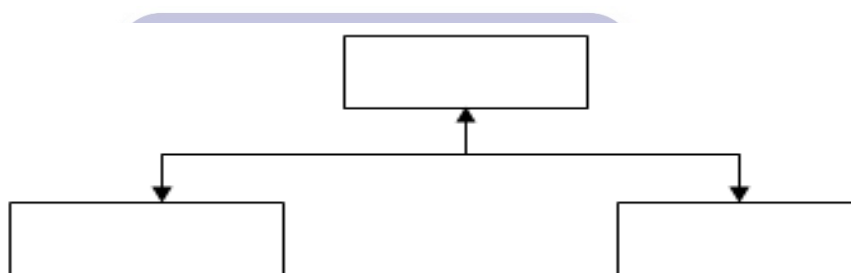


Sumber: Binanto dalam (Eka dan Arviana, 2018)

Gambar II.2
Struktur Navigasi *Linear*

2. *Hierarki*

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

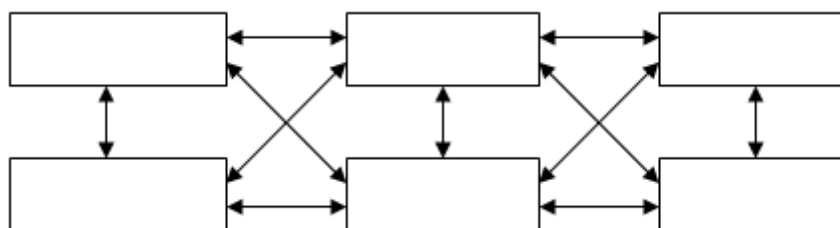


Sumber: Binanto dalam (Eka dan Arviana, 2018)

Gambar II.3
Struktur Navigasi *Hierarki*

3. *Nonlinear*

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

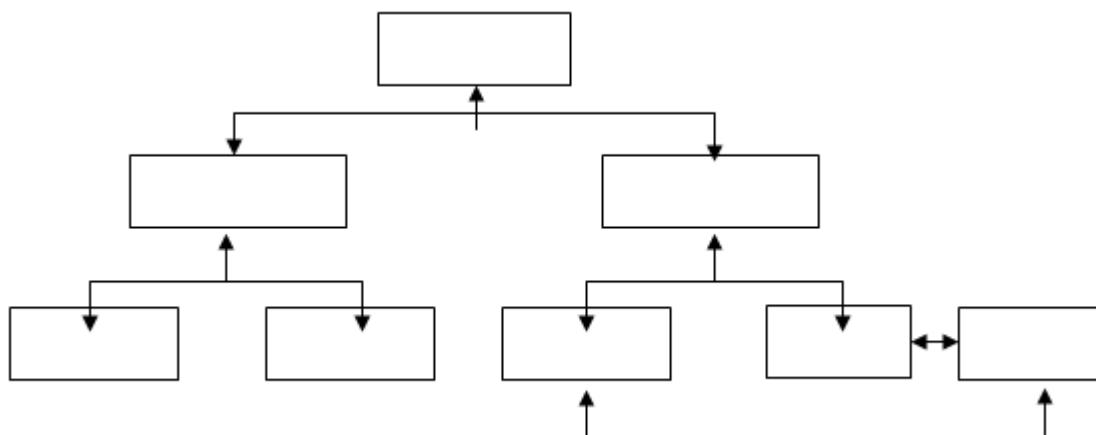


Sumber: Binanto dalam (Eka dan Arviana, 2018)

Gambar II.4
Struktur Navigasi *Nonlinear*

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.



Sumber: Binanto dalam (Eka dan Arviana, 2018)

Gambar II.5
Struktur Navigasi Komposit

B. Sublime Text

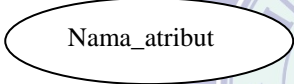
Menurut (Supono dan Putratama, 2018) “*Sublime Text* merupakan peangkat lunak *text editor* yang digunakan untuk membuat atau mengedit suatu aplikasi. *Sublime Text* mempunyai fitur *plugin* tambahan yang memudahkan programmer.”

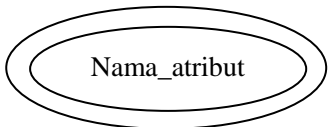
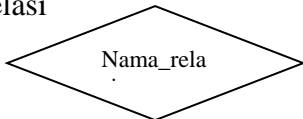
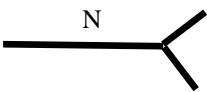
C. ERD (*Entity Relationship Diagram*)

Menurut (Shalahuddin, M dan Sukamto, 2018:50) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional”. Jika menggunakan OODBMS maka perancangan ERD tidak perlu dilakukan.

ERD memiliki beberapa aliran notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen.

Tabel II.1.
Simbol ERD Dengan Notasi Chen

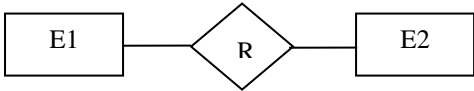
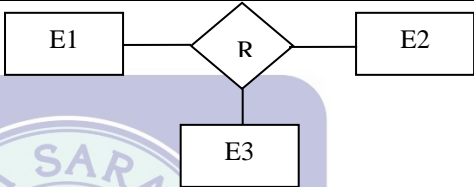
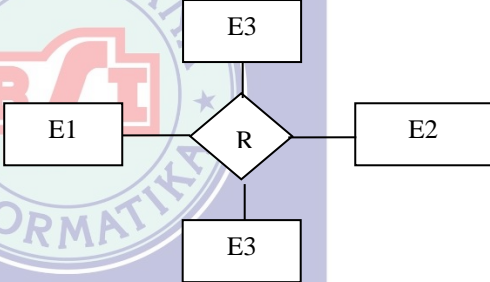
Simbol	Deskripsi
Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, biasanya berupa id, Kunci primer dapat lebih dari satu kolom, asalkan kombinasi dan beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)

Simbol	Deskripsi
Atribut multivalai 	<i>Field</i> atau kolom data yang butuh disimpan dalam satu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja
Asosiasi/ <i>association</i> 	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antar entitas satu dengan entitas yang lain disebut kardinalitas.

Sumber : (Shalahuddin, M dan Sukamto, 2018:50)

Menurut (Shalahuddin, M dan Sukamto, 2018:52) bentuk hubungan relasi dalam ERD sebagai berikut:

Tabel II.2.
Hubungan Relasi dalam ERD

Simbol	Deskripsi
<i>Binary</i>	
<i>Tenary</i>	
<i>N-ary</i>	

Sumber :(Shalahuddin, M dan Sukamto, 2018:52)

LRS (*Logical Record Structure*)

Menurut Frieyadie dalam (Mubarok dan Apriyanta, 2017) menjelaskan bahwa “LRS Merupakan hasil dari pemodelan Entity Relationship (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”.

D. Implementasi dan Penguji Web

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis. Sedangkan untuk menjamin kualitas suatu *web* dibutuhkan adanya pengujian. Pengujian ini juga dimaksudkan untuk mengidentifikasi cacat dan masalah yang ada di *web* tersebut. Berikut penjelasan mengenai dua hal tersebut menurut para ahli:

1. Implementasi

Menurut (Mardiani, 2014) "Tujuan implementasi adalah untuk mengkonfirmasi modul program perancangan pada para pelaku sistem sehingga user dapat memberikan masukan kepada pengembang sistem".

2. Pengujian (*Black Box Testing*)

Menurut (Rizky, 2017:353) "*Black Box testing* adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya tapi cukup dikenal proses testing dibagian luar".