

BAB II

LANDASAN TEORI

2.1. Konsep Dasar *Web*

Teknologi informasi saat ini sangat berpengaruh dalam bidang bisnis. Di zaman sekarang sepertinya tinggal sedikit manusia yang tidak mengenal internet. Internet telah menjadi, kebutuhan utama bagi semua orang karena di dalamnya kita bisa mendapatkan berbagai informasi dari seluruh dunia.

2.1.1. *Website*

Menurut Ardhana dalam (Zamaludin, Isyal, Wina Yusnaeni dan Silvy Amelia, 2016) “*World Wide Web* atau lebih sering dikenal sebagai *Web* adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui *internet*).” Dalam sebuah website terdapat beberapa komponen pendukung agar sebuah website dapat diakses oleh pemakai, diantaranya:

1. Internet

Menurut Winarno dan Zaki dalam (Zamaludin, Isyal, Wina Yusnaeni dan Silvy Amelia, 2016) “Internet adalah singkatan dari kata *Interconnected Networking*. *Networking* artinya jaringan, sedang *Interconnected* berarti Saling berkaitan/terkoneksi. Sehingga internet adalah jaringan komputer yang saling terkoneksi.”

2. *Web Browser*

Menurut Winarno dan Utomo dalam (Prayitno, Agus dan Yulia Safitri, 2015) “*Web browser* adalah alat yang digunakan untuk melihat halaman *web*”.

Software ini kini telah dikembangkan dengan menggunakan *user interface* grafis, sehingga pemakai dapat dengan melakukan *'point and click'* untuk pindah antar dokumen.

3. *Web Server*

Menurut Sandi dalam (Zamaludin, Isyal, Wina Yusnaeni dan Silvy Amelia, 2016) menjelaskan *web server* adalah aplikasi yang menjadi tulang belakang dari *World Wide Web* (WWW). *Web server* menunggu permintaan dari *client* yang menggunakan *web browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, dan aplikasi *web browser* lainnya.

2.1.2. Bahasa Pemrograman

1. PHP (*Personal Home Page*)

Menurut Sidik dalam (Zamaludin, Isyal, Wina Yusnaeni dan Silvy Amelia, 2016) “PHP merupakan Bahasa pemrograman yang paling sering digunakan oleh *programming* web karena merupakan pemrograman *open source*, sehingga para *programming* tidak perlu membeli lisensi untuk membuat aplikasi web”. Rasmus Lerdoft merupakan orang yang membuat PHP pada tahun 1995. Pada waktu itu, nama PHP adalah FI (*Form Interpreted*) yang merupakan sekumpulan *script*, digunakan untuk mengolah data *form* dari web. Pada perkembangan berikutnya, Rasmus akhirnya melepas kode sumber tersebut dan diberi nama PHP (*Personal Home Page*). Dengan berubahnya kode program menjadi *open source*, maka banyak *programming* tertarik dalam perkembangan PHP.

2. HTML (*Hypertext Markup Language*)

Menurut Sidik dalam (Zamaludin, Isyal, Wina Yusnaeni dan Silvy Amelia, 2016) “HTML merupakan kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah *file* teks murni yang dapat dibuat dengan editor teks sembarang”

3. CSS (*Cascading style Sheet*)

Menurut Aditama dalam (Zamaludin, Isyal, Wina Yusnaeni dan Silvy Amelia, 2016) “CSS merupakan salah satu bahasa pemrograman *web* yang bertujuan untuk membuat *web* kita menjadi lebih menarik dan tersruktur, dalam CSS kita bisa merubah warna tabel, besar *font*, atau tata letak menu yang terkendali dari CSS sehingga semua jendela *web* yang berkaitan dengan perubahan tersebut secara otomatis dapat berubah, dengan CSS kita tidak perlu membuat *style* pada setiap *file PHP*, karena cukup dengan satu *file CSS* kita telah bisa mengontrol semua *style* yang kita inginkan pada setiap *file PHP* yang akan ditampilkan nanti pada *web browser*nya.”

4. *Jquery*

Menurut Sugiri dan Kurniawan dalam (Fatmawati, 2016) “*Jquery* merupakan salah satu teknik atau kumpulan *library javascript* yang sangat terkenal animasinya. *jquery* adalah *javascript library*, *jquery* mempunyai semboyan “*write, less, do more*”. *Jquery* dirancang untuk memperingkas kode-kode *javascript*. *Jquery* adalah *javascript* yang cepat dan ringan untuk menangani dokumen HTML, menangani *event*, membuat animasi dan interaksi *ajax*. *Jquery* dirancang untuk mengubah cara anda menulis *javascript*”

5. MySQL (*My Structure Query Language*)

Menurut Anhar dalam (Fatmawati, 2016) “Sebuah *website* yang dinamis membutuhkan tempat penyimpanan data agar pengunjung dapat memberi komentar, saran, dan masukan atas *website* yang dibuat. Tempat penyimpanan data berupa

informasi dalam sebuah tabel disebut dengan *database*. Program yang digunakan untuk mengolah dan mengelola *database* adalah MySQL yang memiliki kumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur, dan menampilkan data. 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 atau *support* dengan *database MySQL*.”

2.1.3. Basis Data

Menurut (Rosa dan Shalahuddin, 2019) “Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS).” Kebutuhan basis data dalam sistem informasi meliputi:

1. Memasukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan.

Tujuan dari dibuatnya tabel-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (record) dimana setiap baris terdiri dari beberapa kolom.

2.1.4. Aplikasi Perancangan Web

1. XAMPP

Menurut Nugroho dalam (Widodo, M. R. R, M, Roziq Zainuddin dan Laura Saraswati Nusantara, 2016), “XAMPP merupakan paket PHP berbasis open source

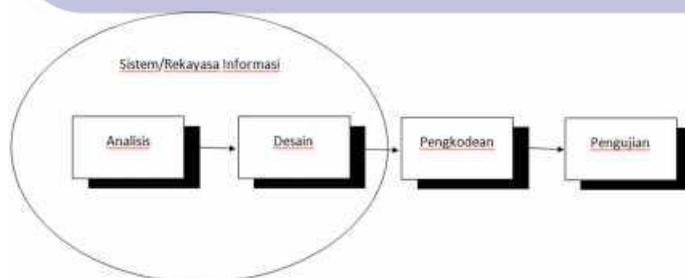
yang berfungsi sebagai server yang berdiri sendiri (*localhost*). Beberapa paket yang telah disediakan adalah Apache, MySQL, PHP, Filezila dan PhpMyAdmin. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan untuk membuat tampilan halaman web dinamis”.

2. Adobe Dreamweaver CS6

Menurut Madcoms dalam (Fatmawati, 2016) “Adobe dreamweaver CS5 adalah sebuah HTML editor professional untuk mendesain sebuah web secara visual dan mengelola situs atau halaman web”. Dreamweaver CS6 tergabung dalam paket Adobe Creative Suite (CS) yang didalamnya terdapat paket desain grafis, video editing, dan pengembangan web aplikasi. Adobe Creative Suite 6 (CS6), dirilis pada tanggal 21 April 2012.

2.1.5. Model Perancangan Perangkat Lunak

Menurut (Rosa dan Shalahuddin, 2019) “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau 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: Rosa dan Shalahuddin (2019)

Gambar II.1

Model *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan 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 bahasa pemrograman. 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.

2.2. Teori Pendukung

Teori pendukung merupakan alat yang digunakan untuk mendukung dan menggambarkan bentuk dari logika model dari suatu sistem dengan menggunakan

simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya.

2.2.1. Struktur Navigasi

Menurut Sukamto, Rosa dan Shalahudin dalam (Zamaludin, Wina Yusnaeni dan Silvy Amelia, 2016) mengemukakan bahwa “Struktur navigasi adalah Navigasi yang ada pada situs *Web* atau aplikasi *Web* menunjukkan sesuatu yang penting dan menjadi kata kunci usability aplikasi.” Ada 4 struktur dasar yang digunakan yaitu *linear*, *non linear*, *hierarki* dan *composite*:

1. *Linear*

Pengguna akan melakukan navigasi secara berurutan, dari *frame* atau *byte* informasi yang satu ke yang lainnya, seperti yang terlihat pada gambar berikut:



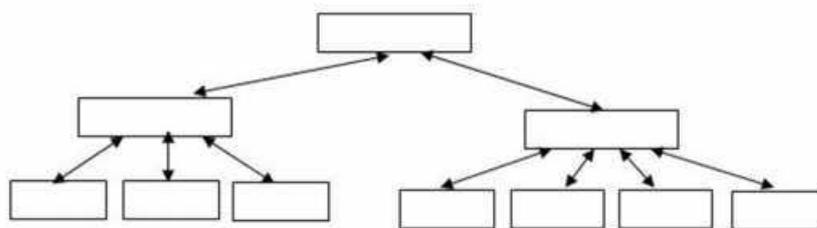
Sumber: (Zamaludin, Wina Yusnaeni dan Silvy Amelia, 2016)

Gambar II.2

Struktur Navigasi *linear*

2. *Hierarki*

Struktur dasar ini disebut juga struktur “*linear* dengan percabangan” karena pengguna melakukan navigasi di sepanjang cabang pohon struktur yang terbentuk oleh logika isi, seperti pada gambar berikut:



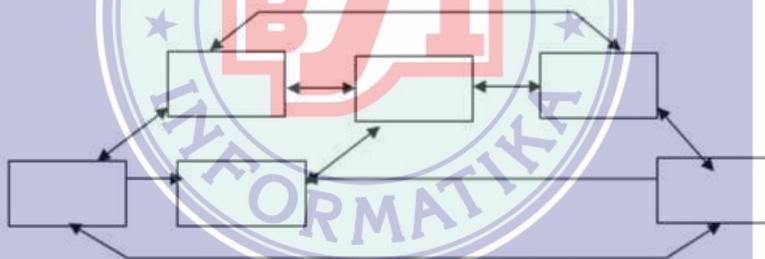
Sumber: (Zamaludin, Wina Yusnaeni dan Silvy Amelia, 2016)

Gambar II.3

Struktur Navigasi *Hierarki*

3. *Nonlinear*

Pengguna akan melakukan navigasi dengan jelas melalui isi proyek dengan titik terkait dengan jalur yang sudah ditentukan sebelumnya, seperti yang terlihat pada gambar berikut:



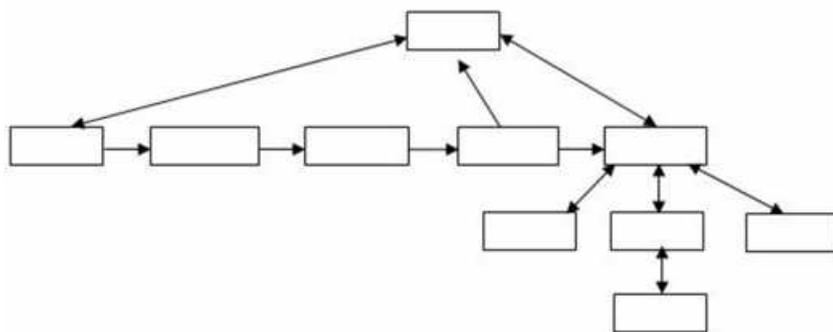
Sumber: (Zamaludin, Wina Yusnaeni dan Silvy Amelia, 2016)

Gambar II.4

Struktur Navigasi *Nonlinear*

4. *Composite*

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



Sumber: (Zamaludin, Wina Yusnaeni dan Silvy Amelia, 2016)

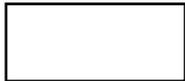
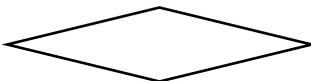
Gambar: II.5

Struktur Navigasi *Composite*

2.2.2. ERD (*Entity Relationship Diagram*)

Menurut Indrajani dalam (Taufik, 2019), Entity Relationship Diagram (ERD) adalah sebuah pendekatan top-bottom dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut entitas dan hubungan antara entitas-entitas tersebut digambarkan dalam suatu model atau diagram.

Menurut (Fathansyah, 2018) “Model *Entity-Relationship* yang berisi komponen-komponen Himpunan Entitas dan Himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari ‘dunia nyata’ yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R)”.

Simbol	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas
	Atribut, yaitu karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas

	Hubungan antara entitas dengan atributnya atau entitas dengan entitas lainnya
--	---

Sumber: (Fathansyah, 2018)

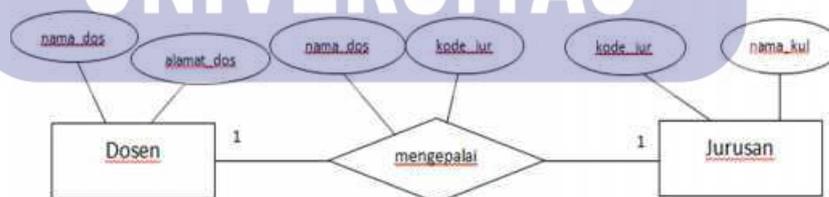
Tabel II.1
Komponen-komponen ERD

1. Derajat *Relationship*

Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak). Berikut adalah contoh penggambaran relasi antar himpunan entitas lengkap dengan kardinalitas relasi dan atribut-atributnya:

1. Relasi satu-ke-satu (one-to-one)

Adanya relasi antara himpunan entitas Dosen dengan himpunan entitas Jurusan. Himpunan relasinya kita beri nama 'Mengepalai'. Pada relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua dosen yang menjadi ketua jurusan). Dan setiap jurusan pasti dikepalai oleh banyak satu orang dosen. Maka penggambarannya adalah:



Sumber: (Fathansyah, 2018)

Gambar II.6.

ERD untuk Relasi Satu Ke Satu

Pada diagram E-R diatas dapat kita lihat bahwa himpunan entitas Dosen dan himpunan entitas Jurusan masing-masing memiliki dua buah atribut saja.

Sementara itu, himpunan relasi Mengepalai juga terdapat dua buah atribut yang secara bersama-sama berfungsi juga sebagai *key* pada himpunan relasi tersebut. Karena kedua atribut/*key* tersebut sebetulnya berasal dari atribut *key* dari masing-masing himpunan entitas yang dihubungkannya, maka keduanya digolongkan sebagai *Key Asing (Foreign-Key)*. Adanya kedua atribut *key* tersebut selanjutnya akan dapat menunjukkan dosen mana yang mengepalai suatu jurusan (menjadi ketua jurusan), atau sebaliknya jurusan mana yang dikepalai seorang dosen.

2. Relasi satu-ke-banyak (one-to-many)

Adanya relasi antara himpunan entitas Dosen dengan himpunan entitas Kuliah. Himpunan relasinya kita beri nama 'Mengajar'. Pada relasi ini, setiap dosen dapat mengajar lebih dari satu mata kuliah, sedang setiap mata kuliah diajar hanya oleh paling banyak satu orang dosen. Maka penggambarannya adalah:



Sumber: Fathansyah (2018)

Gambar II.7.

ERD untuk Relasi satu Ke Banyak

Key Asing (Foreign-Key) dari himpunan relasi Mengajar diatas adalah *nama_dos* dan *kode_kul*, yang masing-masing berasal dari himpunan entitas Dosen dan himpunan entitas Kuliah. Tetapi di samping dari kedua atribut *key* tersebut, ada pula dua atribut tambahan yang tidak berasal dari salah satu

himpunan entitas yang dihubungkannya. Hal ini memang dimungkinkan dan bahkan umum terjadi. Dengan adanya keempat atribut tersebut pada himpunan relasi Mengajar, maka dapat kita ketahui jadwal pelaksanaan setiap mata kuliah beserta dosen yang mengajarkannya.

3. Relasi banyak-ke-banyak (*many-to-many*)

Adanya relasi antara himpunan entitas Mahasiswa dengan himpunan entitas Kuliah. Himpunan relasinya kita beri nama 'Mempelajari'. Pada relasi ini, setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Maka penggambarannya adalah:



Sumber: (Fathansyah, 2018)

Gambar II.8.

ERD untuk Relasi Banyak Ke Banyak

Keberadaan himpunan relasi Mempelajari diatas akan memiliki dua fungsi, yaitu untuk menunjukkan mata kuliah mana saja yang diambil oleh seorang mahasiswa (atau mahasiswa mana saja yang mengambil mata kuliah tertentu) dan indeks nilai yang diperoleh seorang mahasiswa untuk mahasiswa tertentu (tentu saja setelah data indeks nilai tersebut disimpan). Dari jumlah potongan diagram E-R diatas tampak sekali adanya kebutuhan bahwa relasi antara satu himpunan entitas dengan himpunan entitas yang lain harus dapat kita ketahui dengan lebih

detail. Keberadaan relasi antar himpunan entitas itu mungkin memang dapat kita duga secara intuitif yang bertitik tolak dari logika yang normal. Misalnya, kita akan lebih mudah dapat menduga atau bahkan memastikan, bahwa antara himpunan entitas Mahasiswa dan himpunan entitas Kuliah terdapat suatu relasi. Demikian juga antara himpunan entitas Dosen dan himpunan entitas Kuliah. Yang perlu diperhatikan pengertian relasi dalam pemodelan data sama sekali berbeda dengan pengertian relasi dalam bahasa pergaulan sehari-hari (yang berarti hubungan antar pertemanan).

2.2.3. LRS (*Logical Record Structure*)

Frieyadie dalam (Zamaludin, Wina Yusnaeni dan Silvy Amelia, 2016) menerangkan bahwa *LRS* adalah singkatan dari *Logical Record Structure*. *LRS* merupakan hasil dari pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antara entitas. Dalam pembuatan LRS terdapat tiga hal yang dapat mempengaruhi, yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

2.2.4. Pengujian Web

Sebuah perangkat lunak perlu dijaga kualitasnya bahwa kualitas bergantung kepada kepuasan pelanggan (*customer*). Pengujian perangkat lunak adalah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*). Verifikasi lebih mengarah terhadap apakah proses pengembangan produk sudah benar dan telah berhasil mengimplementasikan fungsi yang benar, sedangkan validasi lebih ke arah hasil produk apakah sudah selesai dengan yang diinginkan. Pengujian untuk validasi adalah sebagai berikut:

1. *Black-Box Testing* (pengujian kotak hitam)

Menurut (Rosa dan Shalahuddin, 2019) "*Black-Box Testing* adalah 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 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 proses *login* maka kasus uji yang dibuat adalah:

- a. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. 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.