

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Menurut Oktavian (2013:17) memberikan batasan bahwa, “*web browser* adalah program komputer yang digunakan untuk membaca HTML, kemudian menerjemahkan dan menampilkan hasilnya secara visual ke layar komputer”. Anda dapat menggunakan salah satu program *web browser*, seperti Mozilla Firefox, Internet Explorer (IE), Opera, Safari, Google Chrome, dan sebagainya.

A. *Website*

1. *Website*

Menurut Adelheid (2015:11) memberikan batasan bahwa, “*website* merupakan komponen atau kumpulan komponen yang terdiri teks, gambar, suara, animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi”.

Berdasarkan kategorinya, *website* digolongkan menjadi 3:

- a. *Website Statis*: *website* yang mempunyai halaman tidak berubah. Artinya adalah untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengedit code yang menjadi struktur dari *website* tersebut.
- b. *Website Dinamis*: *website* yang secara struktur diperuntukkan untuk *update* sesering mungkin. Biasanya selain utama yang bisa diakses oleh *user* pada umumnya, juga disediakan halaman *backend* untuk mengedit konten dari *website*. Contoh umum mengenai *website* dinamis adalah web

berita atau web portal yang di dalamnya terdapat fasilitas berita, *polling*, dan sebagainya.

- c. *Website* Interaktif: web yang saat ini memang sedang *booming*. Salah satu contoh *website* interaktif adalah blog dan forum. Di *website* ini *user* bisa berinteraksi dan beradu argumen mengenai apa yang jadi pemikiran mereka. Biasanya *website* seperti memiliki *moderator* untuk mengatur supaya topik yang diperbincangkan tidak keluar jalur.

2. Internet

Menurut Husda dan Wangdra (2016:127) memberikan batasan bahwa, “internet merupakan sistem global jaringan komputer yang berhubungan menggunakan standar *Internet Protocol Suite*(TCP / IP) untuk melayani miliaran pengguna di seluruh dunia”. Jaringan tersebut terdiri dari jutaan jaringan pribadi, umum, akademik, bisnis, dan jaringan pemerintah, dari lokal ke lingkup global, yang dihubungkan oleh sebuah kode array yang luas dari teknologi jaringan elektronik, nirkabel dan optik.

B. Bahasa Pemrograman

1. HTML (*HyperText Markup Language*)

Menurut Oktavian (2013:17) memberikan batasan bahwa, “HTML adalah suatu bahasa yang dikenali oleh *webbrowser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*)”. Sedangkan *webbrowser* adalah program komputer yang digunakan untuk membaca HTML, kemudian menerjemahkan dan menampilkan hasilnya secara visual ke layar komputer. Anda dapat menggunakan salah satu program

webbrowser, seperti: *Mozilla Firefox*, *Internet Explorer (IE)*, *Opera*, *Safari*, *Google Chrome*, dan sebagainya.

2. PHP (*HyperText Preprocessor*)

Menurut Sidik (2017:4) memberikan batasan bahwa:

PHP merupakan secara umum dikenal sebagai bahasa pemrograman *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.

Dengan menggunakan PHP maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.

3. CSS (*Cascading Style Sheet*)

Menurut Wahana Komputer (2015:2) memberikan batasan bahwa, “CSS atau *Cascading Style Sheet* adalah sekumpulan kode pemrograman web yang berfungsi untuk mengendalikan beberapa komponen di dalam web sehingga menjadi tampak seragam, berstruktur, dan teratur”. Selain itu, CSS juga dapat diartikan sebagai kumpulan kode yang digunakan untuk mendefinisikan desain dari bahasa *markup*. Salah satu bahasa *markup* adalah HTML.

C. Basis Data

1. Database

Menurut Enterprise (2014:1) memberikan batasan bahwa, “*database* adalah suatu aplikasi yang menyimpan sekumpulan data”. Setiap *database* mempunyai API tertentu untuk membuat, mengakses, mengaturn mencari, dan menyalin data yang ada di dalamnya.

2. MySQL

Menurut Enterprise (2014:2) mengungkapkan bahwa, “MySQL adalah

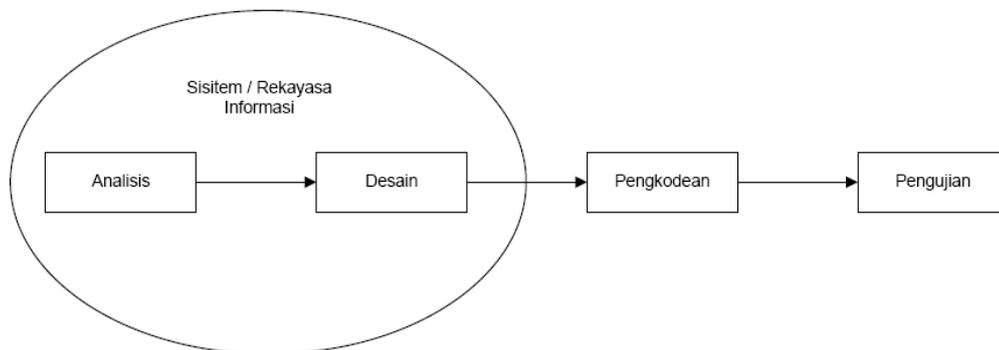
RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan”. MySQL dikembangkan oleh MYSQL AB Swedia.

3. XAMPP

Menurut Winarno, dkk. (2014:1) mengungkapkan bahwa, “XAMPP adalah *software web server* yang bisa dipakai untuk mengakomodasi sistem operasi yang Anda pakai (X), *Apache* (A), MySQL (M), PHP (P) dan *Perl* (P)”. *Perl* adalah bahasa pemrograman lain yang sering dipakai untuk komunikasi antar bahasa pemrograman.

D. Pengembangan Perangkat Lunak Model *Waterfall*

Menurut Rosa dan Shalahuddin (2016:28-30) mengemukakan 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 alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah model gambar air terjun:



Sumber : Rosa dan Shalahuddin (2016:29)

Gambar II.1.
Ilustrasi Model *Waterfall*

Dengan keterangan sebagai berikut:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasi 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 ditranlasikan 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.

2.2. Teori Pendukung

Perancangan program berbasis *website* ini juga tidak terlepas dari teori-teori pendukung, penggunaan teori-teori ini juga dapat mempermudah dalam merencanakan prosedur-prosedur yang akan digunakan oleh program tersebut. Pada perancangan program berbasis *website* ini penulis menggunakan teori pendukung, diantaranya:

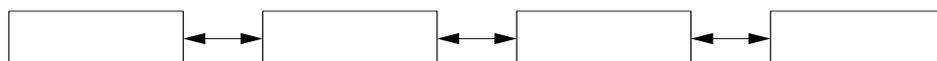
A. Struktur Navigasi

Menurut Andriansyah (2016:61) memberikan batasan bahwa, “struktur navigasi dapat diartikan sebagai alur dari suatu program yang menggambarkan rancangan hubungan antar area yang berbeda sehingga memudahkan proses pengorganisasian seluruh elemen *website*”.

Menurut Binanto dalam Andriansyah (2016:61), ada empat macam bentuk dasar navigasi, yaitu:

1. Struktur Navigasi Berurut (*Linier*)

Struktur navigasi *linier* hanya mempunyai satu rangkaian cerita yang berurut yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya, pengguna akan melakukan navigasi secara berurutan, dalam frame atau byte informasi satu ke yang lainnya.

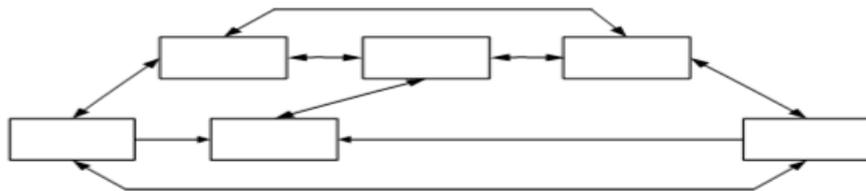


Sumber : Andriansyah (2016:62)

Gambar II.2.
Struktur Navigasi *Linier*

2. Struktur Navigasi Tidak Berurut (*Non-Linier*)

Struktur Navigasi *Linier* merupakan pengembangan dari struktur navigasi *linier*. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur *non-linier* ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan *non-linier* ini walaupun terdapat percabangan tetap tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *Slave Page*, pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat jalur yang sudah ditentukan sebelumnya.

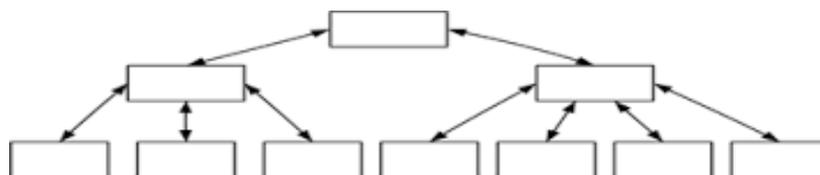


Sumber : Andriansyah (2016:62)

Gambar II.3.
Struktur Navigasi *Non-Linier*

3. Struktur Navigasi Hirarki

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

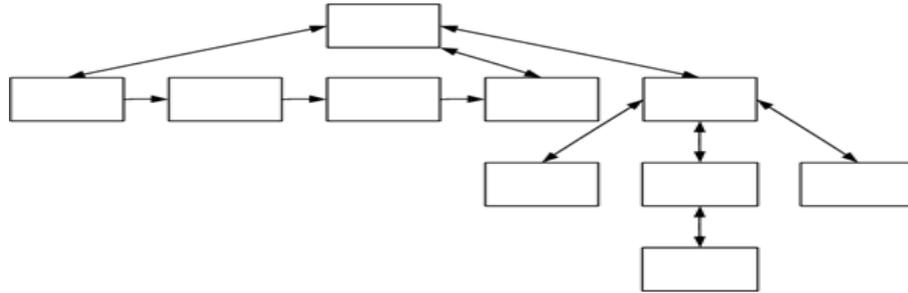


Sumber : Andriansyah (2016:62)

Gambar II.4.
Struktur Navigasi Hirarki

4. Struktur Navigasi Campuran (*Composite*)

Struktur navigasi jenis ini pengguna akan melakukan navigasi dengan bebas (secara *non-linier*), tetapi terkadang dibatasi presentasi *linierfilm* atau informasi penting dan pada data yang paling terorganisasi secara logis pada suatu hirarki.



Sumber : Andriansyah (2016:63)

Gambar II.5.
Struktur Navigasi Campuran

B. ERD (*Entity Relationship Diagram*)

Menurut Husda dan Wangdra (2016:112) memberikan batasan bahwa, “ERD adalah suatu pemodelan dari basis data relational yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang berhubungan antara satu dengan yang lainnya”. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya.

1. Komponen ERD

Simbol-simbol yang digunakan

Menurut Husda dan Wangdra (2016:113), simbol-simbol yang digunakan dalam ERD adalah:

- a. Entitas adalah suatu objek yang sangat dapat diidentifikasi dalam lingkungan pemakai.
- b. Relasi adalah menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
- c. Atribut berfungsi mendiskripsikan karakter entitas (atribut yang berfungsi sebagai *key* diberi garis bawah).
- d. Alur sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

2. Kardinalitas

Menurut Husda dan Wangdra (2016:113) memberikan batasan bahwa, “Kardinalitas (*cardinality*) adalah jumlah minimum dan maksimum kemunculan satu entitas yang mungkin dihubungkan dengan kemunculan tunggal dari entitas lain”. Terdapat tiga jenis kardinalitas:

a. Satu-ke-satu (1:1)

Tingkat hubungan dinyatakan satu-ke-satu jika suatu kejadian pada entitas yang pertama hanya mempunyai satu hubungan dengan satu kejadian yang ada pada entitas kedua, dan sebaliknya, satu kejadian pada entitas yang kedua hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

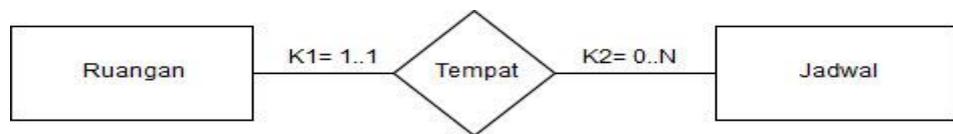


Sumber: Hidayatullah dan Kawistara (2017:147)

Gambar II.6.
Kardinalitas 1:1

b. Satu-ke-banyak (1:N atau N:1)

Tingkat hubungan satu-ke-banyak adalah sama dengan banyak pada satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua, sebaliknya untuk satu kejadian pada entitas yang kedua hanya bisa mempunyai satu hubungan pada entitas yang pertama.



Sumber: Hidayatullah dan Kawistara (2017:147)

Gambar II.7.
Kardinalitas 1:N

c. Banyak-ke-banyak (N:N atau M:N)

Tingkat hubungan banyak-pada-banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi entitas yang kedua.



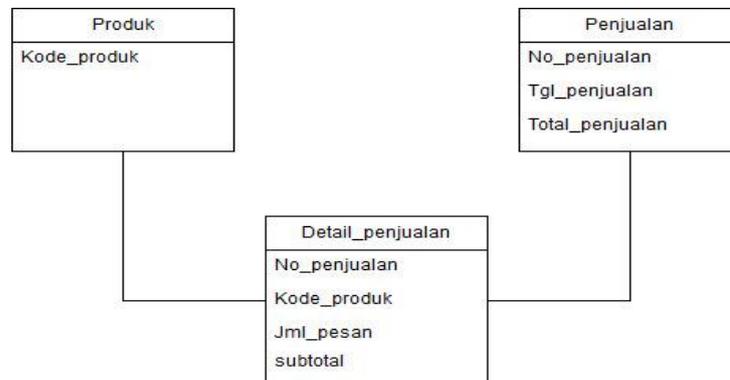
Sumber: Hidayatullah dan Kawistara (2017:147)

Gambar II.8.
Kardinalitas N:N

3. LRS (*Logical Record Structure*)

Menurut Andriansyah (2016:53) memberikan batasan bahwa, “LRS merupakan transformasi dari penggambaran ERD dalam bentuk yang lebih

“jelas dan mudah untuk dipahami”. Penggambaran LRS hampir mirip dengan penggambaran normalisasi *file*, hanya saja tidak digambarkan simbol asterix (*) sebagai simbol *primary key* (kunci utama) dan *foreign key* (kunci tamu).



Sumber : Andriansyah (2016:53)

Gambar II.9.
Rancangan LRS Produk dan Penjualan

C. Pengujian Kotak Hitam(*Black-Box Testing*)

Menurut Rosa dan Shalahuddin (2016:275) memberikan batasan 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. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalnya untuk kasus proses login maka kasus yang dibuat adalah:

1. Jika *user* memasukkan 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.