

BAB II

LANDASAN TEORI

2.1. Kosep Dasar Web

Menurut Azis (2013:25) mengemukakan bahwa “*World Wide Web (WWW)* atau yang biasa kita sebut web merupakan sistem dalam internet yang memiliki fasilitas pencarian dan pemberian informasi yang cepat dengan menggunakan teknologi *hypertext*.”

2.1.1. Website

Seperti yang kita ketahui bahwa website saat ini menjadi tempat yang paling sering atau digunakan dalam kehidupan sehari-hari, baik untuk melihat informasi, membaca informasi, ataupun untuk mendengarkan informasi. Menurut Azis (2013:75), “ Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi internet”.

Dari penjelasan diatas dapat disimpulkan bahwa selama kita terkoneksi internet, kita dapat mencari, mendapatkan informasi kapan saja, dimana saja dan dari mana saja diseluruh dunia.

1. Internet

Menurut Azis, (2013:6), “Internet dapat diartikan sebagai jaringan komputer luas dan besar yang mendunia, yaitu menghubungkan pemakai komputer dari suatu negara ke negara lain di seluruh dunia, dimana di dalamnya terdapat berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif. Internet merupakan sebuah jaringan (Internet Protokol) yang terdiri dari beberapa komputer yang sudah terkoneksi ke dalam jaringan global”.

2. Web Browser

Menurut Azis, (2013:19), ” Penjelajah web (bahasa Inggris: *web browser*), disebut juga sebagai peramban, adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh server web”.

3. Web Server

Menurut Hidayatullah & Kawistara (2016:123), “Web Server adalah tempat dimana anda menyimpan aplikasi web anda kemudian mengaksesnya dengan internet”.

2.1.2. Bahasa Pemrograman

1. HTML

Menurut Ariono (2018:10), “HTML adalah bahasa dasar untuk menampilkan halaman web pada web browser”.

2. PHP

Menurut Subagia (2016:3), “PHP merupakan sebuah bahasa pemrograman yang berjalan dalam sebuah web server (server side)”.

3. CSS

Menurut Ariono (2018:58), “CSS adalah kependekan dari Cascading Style Sheet, berfungsi untuk mempercantik halaman HTML atau menentukan bagaimana elemen HTML ditampilkan, seperti menentukan posisi, merubah warna text atau background dan lain sebagainya.”

4. JavaScript

Menurut Ariono (2018:117), “Javascript adalah bahasa pemrograman web yang berjalan disisi Client/Browser. Javascript biasa digunakan untuk memanipulasi



element-element HTML dan menambahkan Style secara otomatis atau lebih sederhananya membuat dokumen HTML menjadi lebih interaktif’.

2.1.3. Basis Data

Menurut Lubis (2016:3), Basis Data adalah “Tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut.

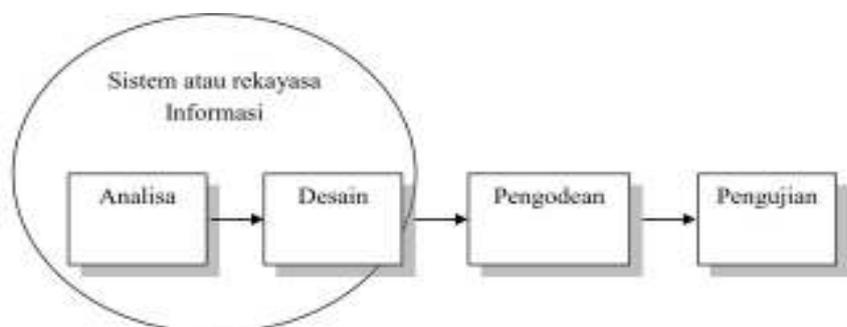
Arti lain dari sistem basis data adalah:

Suatu sistem penyusunan dan pengelolaan record-record dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data secara lengkap pada sebuah organisasi/perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan”.

2.1.4. Model Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak merupakan salah satu dari tahap rancangan aplikasi yang detail dari siklus hidup pengembangan aplikasi. Metode *waterfall* ialah yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*Support*) menurut Sukamto dan Shalahuddin (2016:28).

Adapun gambar model air terjun (*Waterfall*) yaitu:



Sumber : Sukamto dan Shalahuddin (2016:28)

Gambar II.1. Model Waterfall

Model *Waterfall* dibagi menjadi beberapa bagian menurut Sukamto dan Shalahuddin (2016:29) yaitu:

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 proses pengodean. Tahap ini menranslasi 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

Desain harus di translasikan 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

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

2.2.1. Struktur Navigasi

Struktur navigasi adalah Navigasi yang ada pada situs Web atau aplikasi Web menunjukkan sesuatu yang penting dan menjadi kata kunci usability aplikasi. (Zamaludin, Yusnaeni, & Amelia, 2016)

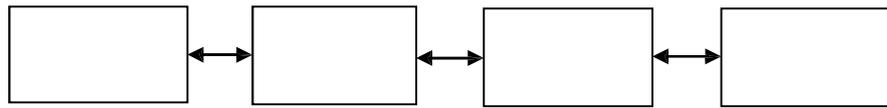
Struktur navigasi dapat digolongkan menurut kebutuhan akan objek, kemudahan pemakaian, keinteraktifitasannya, dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan situs web.

Bentuk dasar dari struktur navigasi adalah sebagai berikut:

1. Struktur Navigasi Linier

Struktur navigasi itu hanya memiliki suatu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang ditampilkan pada struktur jenis ini adalah satu halaman sebelum atau satu haman sesudahnya,

tidak dapat dua halaman sebelumnya atau halaman sesudahnya.

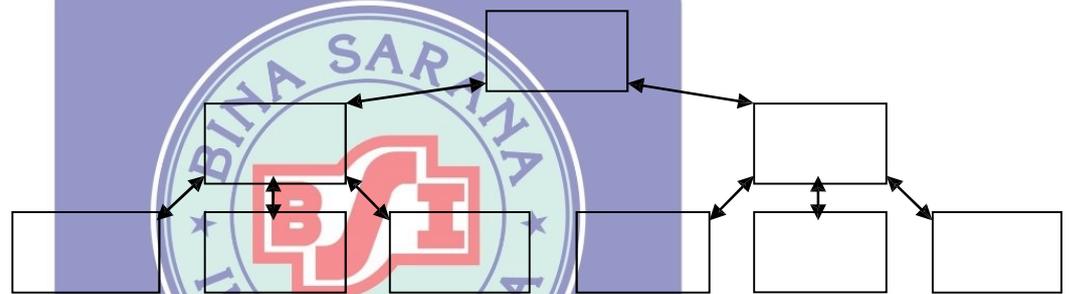


Sumber : Zamaludin et al. (2016:21)

Gambar II.2. Struktur Navigasi Linier

2. Struktur Navigasi Hierarkis

Struktur Navigasi ini disebut juga struktur linear dengan percabangan karena pengguna melakukan navigasi sepanjang cabang struktur yang terbentuk oleh logika isi.

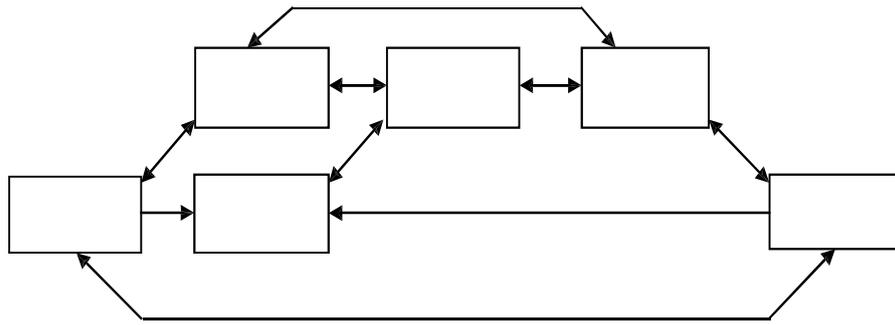


Sumber : Zamaludin et al. (2016:21)

Gambar II.3. Struktur Navigasi Hierarkis

3. Struktur Navigasi Non Linier

Struktur ini disebut juga struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Walaupun terdapat percabangan, tetapi setiap tampilan mempunyai kedudukan yang sama.

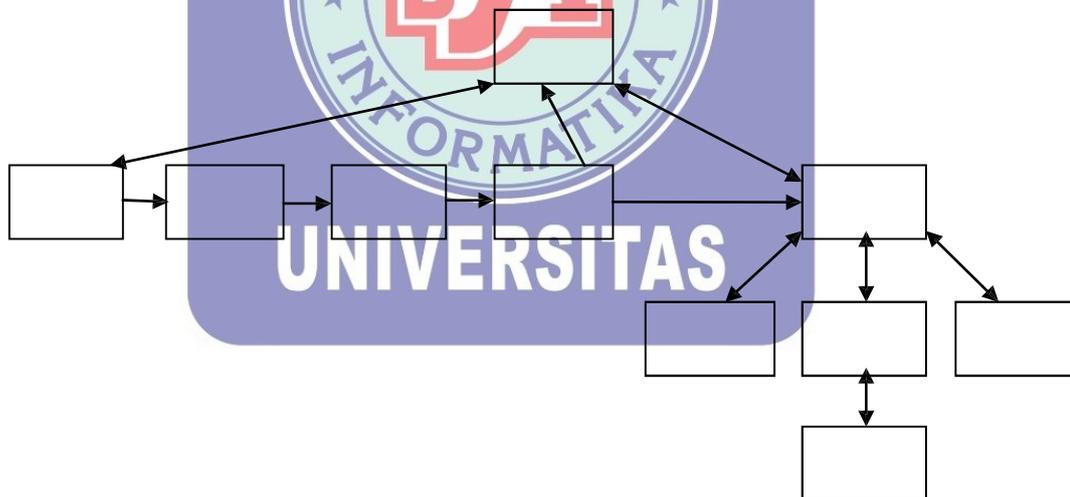


Sumber : Zamaludin et al. (2016:21)

Gambar II.4. Struktur Navigasi Non Linier

4. Struktur Navigasi Komposit atau Campuran

Struktur ini merupakan gabungan dari ketiga struktur navigasi sebelumnya. Struktur navigasi ini juga biasa disebut navigasi bebas. Struktur ini banyak digunakan dalam pembuatan aplikasi multimedia sehingga dapat memberikan keinteraksian yang lebih tinggi.



Sumber : Zamaludin et al. (2016:22)

Gambar II.5. Struktur Navigasi Komposit atau Campuran

2.2.2. Enterprise Relationship Diagram (ERD)

Menurut Ladjamudin (2013:143), memberikan batasan bahwa: “Diagram hubungan atau yang lebih dikenal dengan sebutan ER diagram adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (storage data) dalam sistem secara abstrak.

Diagram hubungan entitas tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data”.

Menurut Sukamto & Shalahuddin, (2016:50), “ERD adalah bentuk paling awal dalam melakukan perencanaan basis data relasional. Jika menggunakan OODBMS maka perancangan ERD tidak perlu dilakukan”.

Berdasarkan uraian diatas dapat disimpulkan bahwa ERD adalah bentuk paling awal dalam melakukan perancangan basis data yang menggunakan susunan data yang disimpan dalam sistem secara abstrak dan menekankan pada struktur dan *relationship* data. Komponen-komponen Diagram Hubungan Entitas:

1. *Entity*

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data.

2. *Relationship*

Pada E-R diagram, *Relationship* digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa digunakan kalimat aktif atau kalimat *pasif*).

3. *Relationship Degree*

Relationship Degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.

4. *Atribut*

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun *Relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa

sebenarnya yang dimaksud entitas maupun Relationship, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan Relationship.

Berdasarkan sifat-sifatnya atribut (Widodo dan Kurniangtyas, 2017:23-24) dapat dibagi menjadi beberapa macam:

a. *Simple atribut*

Adalah sebuah atribut yang tidak dapat dipecah lagi menjadi komponen yang lebih kecil. Contoh: nama-pelanggan, kota-pelanggan.

b. *Composite atribut*

Adalah atribut yang dapat dibagi lagi menjadi yang lebih kecil. Namun masih memiliki satu kesatuan arti. Contoh: alamat-pelanggan, jalan, kota, negara, kodepos.

c. *Singlevalued atribut*

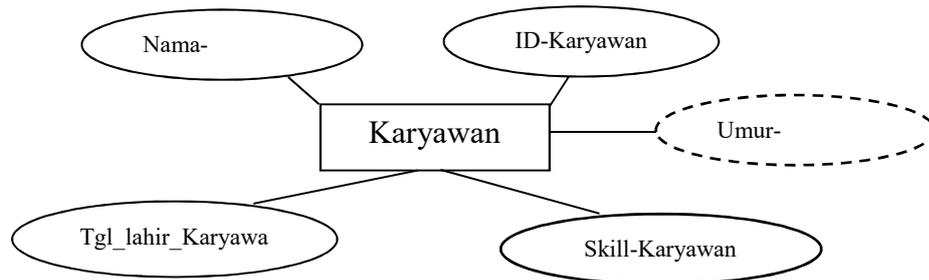
Adalah atribut yang hanya memiliki satu nilai saja. Contoh: ID-Karyawan. Atribut jenis ini disimbolkan dengan *elips* dengan garis tunggal.

d. *Multivalued atribut*

Adalah atribut yang memungkinkan memiliki nilai lebih dari satu. Contohnya: skill-karyawan, hoby-karyawan, NoTelpKaryawan. Simbol dari atribut multivalued adalah *elips* dengan garis ganda.

e. *Derived atribut*

Adalah atribut yang nilainya diperoleh dari menurunkan nilai dari hubungan atribut lainnya. contohnya: Umur-Karyawan (umur bisa dihitung dari Tanggal_lahir_karyawan). Atribut ini turunan ini disimbolkan dengan *elips* yang memiliki garis putus-putus.



Sumber : Widodo & Kurnianingtyas (2017:25)

Gambar II.6. Contoh Singlevalued atributs, Multivalued atributs, Derived atribut

5. Kardinalitas (*Cardinality*)

Menurut Yanto (2016:40), “Derajat kardinalitas merupakan penjabaran dari hubungan antara entitas”. Derajat kardinalitas di bagi menjadi 3 bagian:

a. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.



Sumber : Yanto (2016:41)

Gambar II.7. Kardinalitas *One to One*

b. *One to Many* atau *Many to One*

Tingkat hubungan satu banyak adalah sama dengan banyak ke satu.



Sumber : Yanto (2016:41)

Gambar II.8. Kardinalitas *One to Many*

c. *Many To Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya.



Sumber : Yanto (2016:41)

Gambar II.9. Kardinalitas *Many to Many*

Menurut Sukamto dan Shalahuddin (2016:50), “ERD memiliki beberapa aliran notasi atau komponen”.

2.2.3. Pengujian Web

Pengujian web dilakukan dengan metode Black Box Testing yang merupakan proses pengujian yang fokus pada proses masukan dan keluaran pada saat sistem informasi dijalankan. Penggunaan metode Black Box Testing pada tahap ini karena bertujuan untuk mengetahui sejauh mana unit program dapat memenuhi kebutuhan (requirement) yang disebutkan dalam analisis pengguna. (Andriani, 2015:29).

Menurut Sukamto dan Shalahuddin (2016:275), “*Black-Box Testing* (Pengujian Kotak Hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Pengujian pada *Black Box* berusaha menemukan kesalahan seperti:

1. Fungsi-fungsi yang tidak benar atau hilang Kesalahan interface.

2. Kesalahan dalam struktur data atau akses database eksternal.
3. Kesalahan kinerja.
4. Inisialisasi dan kesalahan terminasi.

Kelebihan *Black Box Testing* ini adalah spesifikasi program dapat ditentukan di awal, dapat digunakan untuk menilai konsistensi program, testing dilakukan berdasarkan spesifikasi, dan tidak perlu melihat kode program secara *detail*. Kekurangan *Black Box Testing* yaitu bila spesifikasi program yang dibuat kurang jelas dan ringkas, maka akan sulit membuat dokumentasi setepat mungkin.

Pengujian *black box* atau kotak hitam dilakukan dengan mencoba semua fungsi yang terdapat pada aplikasi yang dibangun. Pengujian ini dilakukan dengan tujuan untuk mengetahui apakah fungsi fungsi tersebut sesuai dengan kebutuhan. Kasus uji yang dibuat untuk melakukan pengujian black box harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login menurut Sukamto dan Shalahudin (2016:276) maka kasus uji 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.

Tabel II.1
Contoh Hasil *Black Box Testing* terhadap *Form Login Admin*

No	Skenario Pengujian	Test Case	Hasil yang di harapkan	Hasil pengujian	Kesimpulan
1.	<i>Username</i> dan <i>password</i> tidak diisi kemudian klik tombol <i>login</i> .	<i>Username:</i> Kosong <i>Password:</i> kosong	Sistem akan Menolak akses <i>user</i> dan menampilkan "Maaf semua	Sesuai harapan	<i>Valid</i>

			<i>Field</i> harus diisi”		
2.	Mengetikan <i>Username</i> diisi dan <i>password</i> tidak diisi atau kosong kemudian klik tombol <i>login</i>	<i>Username:</i> Lehal <i>Password:</i> Kosong	Sistem akan menolak akses <i>user</i> dan menampilkan “Maaf semua <i>field</i> harus diisi”	Sesuai harapan	<i>Valid</i>
3	<i>Username</i> tidak diisi dan <i>password</i> diisi kemudian klik tombol <i>login</i>	<i>Username:</i> Kosong <i>Password:</i> Lehal	Sistem akan menolak akses <i>user</i> dan menampilkan “Maaf semua <i>field</i> harus diisi”	Sesuai harapan	<i>Valid</i>
4.	Mengetikan salah satu kondisi salah satu kondisi pada <i>username</i> dan <i>password</i> kemudian klik tombol <i>login</i>	<i>Username:</i> Lehal (benar) <i>Password:</i> Lehal123 (salah)	Sistem akan menolak akses <i>user</i> dan menampilkan “Maaf <i>username</i> dan <i>password</i> tidak valid”	Sesuai harapan	<i>Valid</i>
5.	Mengetikan <i>username</i> dan <i>password</i> dengan data yang benar kemudian klik tombol <i>login</i>	<i>Username:</i> Lehal (benar) <i>Password:</i> Lehal194 (benar)	Sistem menerima akses <i>login</i> dan kemudian langsung menampilkan menu utama	Sesuai harapan	<i>Valid</i>

Sumber : Zamaludin et al., (2016:25)