

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Web dikenal dengan istilah WWW (*World Wide Web*) yang merupakan layanan *internet* yang paling populer dan dikenal secara luas. WWW sebagai halaman-halaman *website* yang dapat saling terkoneksi antara satu dengan yang lainnya (*hyperlink*).

Dalam pengertian WWW (*World Wide Web*) menurut Arief (2011:8) menerangkan bahwa “*World Wide Web* atau yang biasa disingkat WWW merupakan kumpulan situs *web* yang dapat diakses di *internet* yang berisikan semua informasi yang dibutuhkan semua pengguna *internet*”.

Sedangkan menurut Kadir dan Triwahyuni (2013:309) bahwa “*World Wide Web* (WWW) atau *Web* merupakan sumber daya *Internet* yang sangat populer dan dapat digunakan untuk memperoleh informasi atau bahkan melakukan transaksi pembelian barang”.

Berdasarkan dari kutipan di atas bahwa WWW (*World Wide Web*) merupakan kumpulan situs *Web* yang berisikan semua informasi serta dapat melakukan transaksi pembelian barang yang dapat diakses oleh semua pengguna *internet*.

2.1.1. Website

Website terdapat beberapa istilah yang diantaranya sebagai berikut:

1. *Internet*

Internet kependekan dari kata *Interconnection-Networking* yang merupakan seluruh jaringan komputer yang saling terhubung untuk melayani seluruh pengguna *internet* di seluruh dunia.

Pengertian *internet* menurut Sibero (2013:10) menjelaskan “*Internet (Interconnected Network)* adalah jaringan komputer yang menghubungkan antara jaringan secara global, *internet* dapat juga disebut jaringan dalam suatu jaringan yang luas”.

Selain itu *internet* menurut Hidayatullah dan Kawistara (2015:1) bahwa “*Internet* adalah jaringan global yang menghubungkan komputer-komputer di seluruh dunia. Dengan *internet*, sebuah komputer bisa mengakses data yang terdapat pada komputer lain di benua yang berbeda”.

Dari uraian di atas dapat dimengerti bahwa *internet* sebagai rangkaian jaringan komputer yang terhubung serta dapat saling berkomunikasi secara global yang menghubungkan dengan seluruh komputer secara luas.

2. *Web Server*

Untuk menghasilkan tampilan *website* pada umumnya, memerlukan *web server* yang berupa *software* yang memberikan layanan berbasis data untuk menerima permintaan dari HTTP atau HTTPS pada sisi *client*.

Menurut Arief (2011:19) mengemukakan bahwa “*Web server* adalah program aplikasi yang memiliki fungsi sebagai tempat menyimpan dokumen-dokumen *web*”.

Sedangkan menurut Sibero (2013:11) “*Web server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak”.

Dalam penjelasan tersebut dapat dipahami *web server* adalah perangkat lunak yang dijalankan di komputer *server* sebagai tempat menyimpan dokumen *web* untuk diakses oleh pengguna.

3. *Web Browser*

Web Browser digunakan untuk menjelajahi *internet*, yang dapat menampilkan informasi berupa teks, gambar, *file* multimedia seperti *audio* dan *video*.

Pengertian *web browser* menurut Badiyanto (2013:23) “*Web browser* adalah sebuah program yang mengakses dan menerjemahkan kode perintah dari dokumen HTML”.

Sedangkan dalam pengertian *web browser* menurut Sibero (2013:12) bahwa “*Web browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*”.

Dari teori di atas *web browser* sebagai program untuk menampilkan halaman *web* dan menyajikan sumber informasi *web*.

2.1.2. *E-Learning*

E-learning merupakan kepanjangan dari *elektronik learning* yang merupakan metode baru dalam proses belajar mengajar dengan menggunakan media elektronik khususnya *internet* sebagai sistem pembelajarannya. *E-learning* mempunyai ciri-ciri, antara lain (Clark & Mayer dalam Hanum, 2013:92), diantaranya:

1. Memiliki konten yang relevan dengan tujuan pembelajaran.
2. Menggunakan metode instruksional, misalnya penyajian contoh dan latihan untuk meningkatkan pembelajaran.
3. Menggunakan elemen-elemen media seperti kata-kata dan gambar-gambar untuk menyampaikan materi pembelajaran.
4. Memungkinkan pembelajaran langsung berpusat pada pengajar (*synchronous e-learning*) atau di desain untuk pembelajaran mandiri (*asynchronous e-learning*).
5. Membangun pemahaman dan keterampilan yang terkait dengan tujuan pembelajaran baik secara perseorangan atau meningkatkan kinerja pembelajaran kelompok.

Berdasarkan dari pengertian *E-Learning* di atas dapat disimpulkan bahwa *E-learning* adalah metode baru dalam proses belajar mengajar dengan menggunakan media elektronik khususnya *internet*.

2.1.3. Basis Data (*Database*)

Basis data atau biasa disebut dengan *database* yang berfungsi untuk menyimpan atau mengolah sebuah data. Maka dapat dikatakan basis data sangat dibutuhkan dalam menampung data yang diolah oleh sebuah perangkat lunak.

Berdasarkan dalam pengertian basis data menurut Rosa dan Shalahuddin (2013:43) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah di olah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media yang menyimpan data agar dapat diakses dengan mudah dan cepat”.

Sedangkan menurut Badiyanto (2013:57) “*Database* bisa dikatakan sebagai suatu kumpulan dari data yang tersimpan dalam tabel dan diatur atau diorganisasikan sehingga data tersebut bisa diambil atau dicari dengan mudah dan efisien”.

Dengan adanya pengertian di atas dapat disimpulkan bahwa basis data adalah media yang menyimpan data dan diatur atau diorganisasikan sehingga data dapat diambil atau dicari dengan cepat jika sewaktu-waktu akan dibutuhkan.

1. MySQL

MySQL sebagai *server database open source* yang digunakan pada aplikasi terutama dalam membuat *web*, MySQL digunakan dalam mengolah data yang terdapat pada *database*.

Menurut Anhar (2010:45) menjelaskan bahwa “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.

Sedangkan menurut Sibero (2013:97) yaitu “MySQL atau dibaca ‘*My Sekuel*’ dengan adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”.

Berdasarkan dari penjelasan tersebut, MySQL menggunakan bahasa SQL yang berfungsi dalam mengolah *database*.

2. *phpMyAdmin*

phpMyAdmin adalah aplikasi *Web* yang dibuat oleh *phpmyadmin.net*. *phpMyAdmin* digunakan untuk administrasi *database* MySQL. Beberapa fitur administrasi *database* pada *phpMyAdmin* yang diantaranya pembuatan

database, mengubah *database*, pembuatan tabel, mengubah tabel, menghapus tabel, menambah data (*Record*), menampilkan data (*Record*), mengubah data (*Record*), menghapus data (*Record*), membuat *view*, menghapus *view*, membuat *index* kolom, menghapus *index* kolom (Sibero, 2013:376). Dengan begitu, *phpMyAdmin* merupakan aplikasi pengolah *database*.

2.1.4. Model Pengembangan *Waterfall*

Model *waterfall* sebagai metode dalam pengembangan sistem ini. Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Dalam metode ini terdapat lima tahapan (Rosa dan Shalahuddin, 2015:28), yaitu:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan untuk menspesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Tahapan ini terfokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahapan ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

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

Tahapan pengujian dilakukan agar mengurangi kesalahan (*error*) untuk memastikan keluaran yang dihasilkan sesuai dengan yang diharapkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk merubah perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Teori Pendukung

Dalam bagian ini menjelaskan tentang teori-teori pendukung yang digunakan dalam proses pembuatan sistem ini.

2.2.1. Aplikasi

Pengertian Aplikasi menurut Jogiyanto dalam Rini dkk (2013:6) mengemukakan “Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga *computer* dapat memproses *input* menjadi *output*.”

Adapun menurut Dhanta dalam Permatasari dan Prehanto (2013:14), “Aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, dan *Microsoft Excel*.”

Terdapat pula pengertian Aplikasi menurut Supriyanto dalam Mustika (2012:201), “Aplikasi adalah program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.”

Memahami tentang adanya pengertian Aplikasi di atas maka penulis berkesimpulan Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

2.2.2. Bahasa Pemrograman

Bahasa pemrograman atau bahasa pemrograman komputer, sebagai instruksi standar dalam memerintah komputer, bahasa pemrograman yang merupakan suatu himpunan dari aturan sintaks yang dipakai untuk mendefinisikan program komputer.

1. Hypertext Markup Language (HTML)

Dalam pembuatan sebuah halaman *web* yang menggunakan bahasa pemrograman HTML untuk menampilkan berbagai informasi.

HTML menurut Oktavian (2013:17) “HTML adalah suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*)”.

Menurut Sibero (2013:19) mengemukakan bahwa “*HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*”.

Selain itu menurut Arief (2011:23) “HTML atau *HyperText Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman *web*”.

Sedangkan dalam pengertian HTML menurut Badiyanto (2013:23): HTML (*Hyper Text Markup Language*) adalah bahasa standar pemrograman untuk membuat halaman *web* yang terdiri dari kode-kode *tag* tertentu, kemudian kode-kode tersebut diterjemahkan oleh *web browser* untuk menampilkan halaman *web* yang terdiri dari berbagai macam format tampilan seperti teks, grafik, animasi *link*, maupun audio-video.

Dari pengertian di atas HTML (*Hypertext Markup Language*) sebagai bahasa pemrograman yang digunakan pada halaman *web* dalam pembuatan dokumen dan aplikasi, sebagai bahasa untuk pertukaran dokumen *web*, dan menampilkan informasi lebih menarik pada halaman *web* yang terdiri dari berbagai macam format tampilan seperti teks, animasi *link*, *audio-video*.

2. *Hypertext Preprocessor* (PHP)

PHP digunakan untuk pemrograman *web* dinamis, yaitu pengguna dapat merubah isi konten dari halaman tertentu.

Menurut Arief (2011:43) mengungkapkan “PHP (*PHP: Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis”.

Sedangkan menurut Oktavian (2013:69) menerangkan bahwa “PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”.

Maka dapat dipahami PHP adalah bahasa *server-side scripting* yang digunakan untuk membuat *web* dinamis dan mengolah data yang selanjutnya dikirimkan ke *web browser* menjadi kode HTML.

3. *Cascading Style Sheet (CSS)*

Cascading Style Sheet (CSS) digunakan pada tahapan pendesainan halaman *web*, sehingga tampilan dari halaman *web* lebih menarik.

Pengertian CSS (*Cascading Style Sheet*) menurut Badiyanto (2013:25): CSS adalah kepanjangan dari *Cascading Style Sheets*. CSS skrip yang berisi rangkaian instruksi yang menentukan suatu teks akan tertampil di halaman *web browser*. Perancangan tampilan *web* dapat dilakukan dengan mendefinisikan *font* (huruf), *color* (warna), *margin* (ukuran), latar belakang (*background*), dan ukuran *font* (*font size*) dan lain-lain. Elemen-elemen seperti *color* (warna), *font* (huruf), *sizes* (ukuran) dan *spacing* (jarak) disebut juga “*style*”.

Sedangkan menurut Sibero (2013:112) menyimpulkan bahwa “*Cascading Style Sheet* dikembangkan untuk menata gaya pengaturan halaman web”.

Menyimpulkan dari pengertian di atas bahwa CSS (*Cascading Style Sheet*) sebagai perancangan tampilan halaman web.

4. *JavaScript*

Javascript berfungsi untuk membuat interaksi antara *user* dengan *web* menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*.

Dalam pengertian *Javascript* menurut Sibero (2013:150) bahwa “*Javascript* adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan pada *web browser*”. *Javascript* adalah bahasa skrip (*Scripting Language*), yaitu kumpulan instruksi perintah yang digunakan untuk mengendalikan beberapa bagian dari sistem operasi.

Sedangkan pengertian *Javascript* menurut Oktavian (2013:51) “*JavaScript* adalah *script* yang digunakan untuk mengatur interaksi antara pengguna dengan *web*”.

Memahami dari pengertian di atas, *Javascript* merupakan suatu instruksi atau perintah yang berjalan pada *web browser* dan berinteraksi dengan pengguna.

5. *JQuery*

JQuery yang merupakan kode *javascript* siap pakai, dan dapat digunakan dalam mengubah tampilan bagian halaman tertentu, mengubah isi halaman, menambah animasi, melakukan load data, dan meringkas sintaks *Javascript*.

Sigit (2011:1) menjelaskan bahwa “*Jquery* adalah *library* atau kumpulan kode *JavaScript* siap pakai. Keunggulan menggunakan *jquery* dibandingkan dengan *javascript* standar, yaitu menyederhanakan kode *javascript* dengan cara memanggil fungsi-fungsi yang disediakan oleh *jquery*”.

Adapun menurut Hakim (2013:3) bahwa “*JQuery* adalah *Javascript Library*/kumpulan kode/fungsi *Javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *Javascript*”.

Dari penjelasan di atas, *JQuery* merupakan kumpulan kode *Javascript* yang dapat disederhanakan dengan cara memanggil fungsi-fungsi yang sudah disediakan oleh *JQuery*, agar dapat mempermudah dalam pembuatan halaman *web* yang menarik.

2.2.3. *Sublime Text 3*

Dalam proses penulisan *coding*, tentunya akan menggunakan salah satu dari sekian banyak *text editor* yang ada. *Text editor* sebagai aplikasi yang digunakan dalam membuat atau mengedit teks.

Sublime text adalah teks editor berbasis *Python*, sebuah teks *editor* yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal dikalangan *developer* (pengembang), penulis dan *desainer*. *Sublime text* mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna

dalam membuat sebuah *web development*. Berikut keunggulan-keunggulan fitur yang dimiliki *Sublime Text 3* (Faridl, 2015:3), adalah:

1. *Multiple Selection*

Multiple Selection memiliki fungsi membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. Dengan meletakkan kursor pada kode yang akan diubah, lalu tekan *Ctrl+klik* atau blok kode yang akan diubah kemudian *Ctrl+D* setelah itu kode dapat diubah secara bersamaan.

2. *Command Pallete*

Command Pallete berfungsi untuk mengakses *file shortcut* dengan mudah. Untuk mencari *file* tersebut, silahkan tekan *Ctrl+Shift+P* dan cari perintah yang diinginkan.

3. *Distraction Free Mode*

Fitur ini berfungsi untuk merubah tampilan layar menjadi penuh dengan menekan *Shift+F11*.

4. *Find in project*

Dapat mencari dan membuka *file* di dalam sebuah *project* dengan mudah dan cepat. Dapat mencari *file* yang diinginkan dengan menekan *Ctrl+P*.

5. *Plugin API Switch*

Mempunyai keunggulan dengan *plugin* yang berbasis *Python Plugin API*. Teks *editor* ini mempunyai banyak *plugin* yang beragam dan memudahkan pengguna dalam mengembangkan *software*.

6. *Drag and Drop*

Dalam teks *editor* ini pengguna dapat menyeret dan melepas *file* teks ke dalam *editor* yang akan membuka *tab* baru secara otomatis.

7. *Split Editing*

Pengguna dapat mengedit *file* secara berdampingan dengan klik *File->New menu into file*.

2.2.4. XAMPP

XAMPP sebagai *server localhost* untuk menerjemahkan bahasa yang ditulis dalam bahasa pemrograman PHP.

Pengertian *XAMPP* menurut Madcoms (2011:13) “*XAMPP* dan *AppServ* adalah salah satu paket *software web server* yang terdiri dari *Apache*, *MySQL*, *Php*, dan *PhpMyAdmin*”.

Sedangkan menurut Arief (2011:44) bahwa “*XAMPP* merupakan aplikasi yang mengintegrasikan beberapa aplikasi utama *web* didalamnya. Dalam *XAMPP* terdapat instalasi modul *PHP*, *MySQL*, *web server Apache*”.

Selain itu menurut Badiyanto (2013:9) mengemukakan bahwa “*XAMPP* adalah perangkat lunak *server* yang menyediakan aplikasi *server Apache*, *PHP*, *MySQL*”.

Berdasarkan dari pengertian diatas *XAMPP* adalah sebuah aplikasi *web server* yang terdiri dari *Apache*, *MySQL database*, dan menerjemahkan bahasa pemrograman *PHP*.

2.2.5. Struktur Navigasi

Dalam pembuatan aplikasi multimedia, struktur navigasi termasuk struktur terpenting dalam tahap perancangan suatu hubungan dan rantai kerja yang menghubungkan dari suatu halaman ke halaman lainnya.

Menurut Binanto dalam Purnomo (2010:269) “Terdapat empat macam struktur navigasi, yaitu *linear navigation* model, *hierarchical* model, *spoke-and-hub* model, dan *full web* model”.

1. *Linear*

Pengguna akan melakukan navigasi secara berurutan, dari *frame* atau *byte* informasi yang satu ke yang lain.

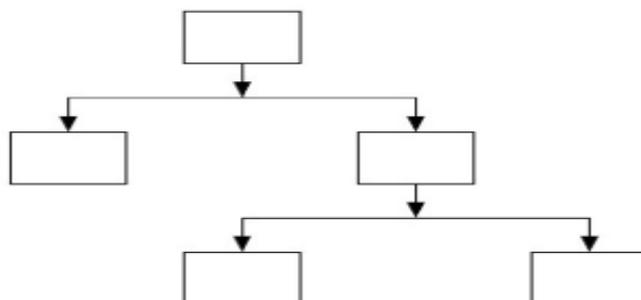


Sumber: Binanto dalam Purnomo (2010:269)

Gambar II.1. Struktur Navigasi *Linear*

2. *Hierarkis*

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

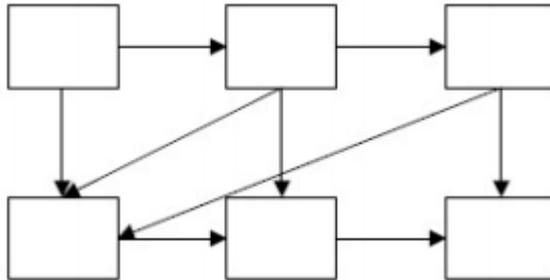


Sumber: Binanto dalam Purnomo (2010:269)

Gambar II.2. Struktur Navigasi *Hierarkis*

3. *Nonlinear*

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

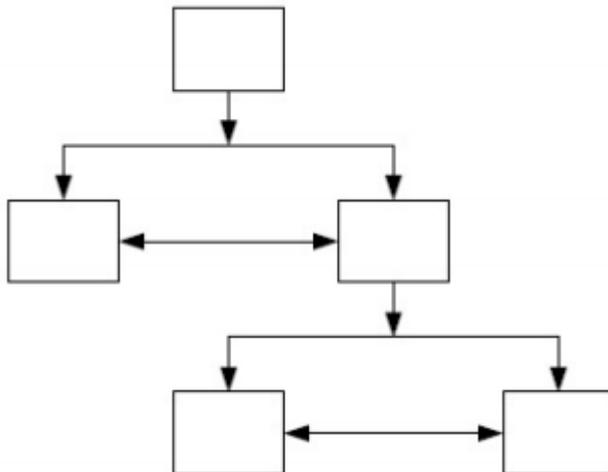


Sumber: Binanto dalam Purnomo (2010:269)

Gambar II.3. Struktur Navigasi *Nonlinear*

4. *Komposit*

Pengguna akan melakukan navigasi dengan bebas (secara *non-linear*), tetapi terkadang dibatasi presentasi *linear film* atau informasi penting dan atau pada data yang paling terorganisasi secara logis pada suatu *hierarki*.



Sumber: Binanto dalam Purnomo (2010:269)

Gambar II.4. Struktur Navigasi *Komposit*

2.2.6. *Entity Relationship Diagram (ERD)*

ERD (*Entity Relationship Diagram*) adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan *field-field* di dalamnya pada suatu *database* sistem. Keterkaitan antar tabel ini bisa disebut dengan relasi. Terdapat tiga buah jenis relasi antar tabel di dalam bagan ERD. Adapun jenis-jenis relasi tersebut (Pratama, 2014:49), sebagai berikut:

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

Relasi ini menggambarkan hubungan satu *field* pada tabel pertama ke satu *field* pada tabel kedua.

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

Menggambarkan hubungan satu *field* pada tabel pertama ke dua atau beberapa buah *field* di tabel kedua.

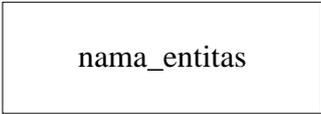
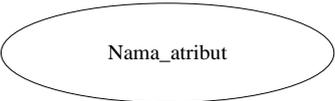
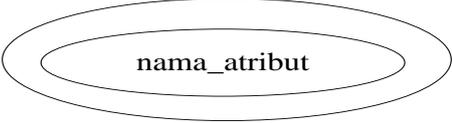
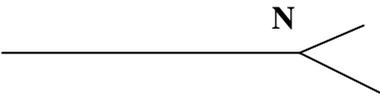
3. *Many to Many* (Banyak ke Banyak)

Satu atau lebih *field* pada tabel pertama dapat dihubungkan ke satu atau lebih ke *field* pada tabel kedua dan membentuk tabel ketiga.

2.2.7. *Komponen ERD (Entity Relationship Diagram)*

Entity Relationship Diagram (ERD) banyak digunakan dalam pemodelan basis data. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Akan tetapi yang banyak digunakan adalah notasi dari Chen. Berikut simbol-simbol atau komponen yang digunakan pada ERD dengan notasi Chen (Rosa dan Shalahuddin, 2013:50):

Tabel II.1. Komponen ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penanaman 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; berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multilicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau disebut dengan One to Many menghubungkan entitas A dan entitas B.

Sumber: Rosa dan Shalahuddin (2013:50)

2.2.8. Logical Record Structure (LRS)

Logical Record Structure (LRS) merupakan hasil transformasi ERD ke LRS yang melalui proses kardinalitas dan menghasilkan atribut-atribut yang saling berelasi Ladjamudin (2013:159).

Selain itu menurut Pradani dkk (2013:99) mengatakan bahwa “LRS adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas”.

Mengambil kesimpulan dari pengertian di atas bahwa LRS (*Logical Record Structure*) merupakan struktur *record* pada tabel yang menghasilkan relasi antar atribut.

2.2.9. Pengujian Unit

Black Box Testing (pengujian kotak hitam) yaitu pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui terhadap fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji dibuat dengan kasus benar dan kasus salah, misalkan kasus proses login (Rosa dan Shalahuddin, 2013:275), 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.

Sedangkan menurut Rizky dalam Rahayu (2011:264) “*blackbox 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 dikenai proses *testing* di bagian luar”.

Klasifikasi *black box testing* mencakup beberapa pengujian menurut Simarmata dalam Rahayu (2010:316), yaitu:

1. Pengujian fungsional (*functional testing*)

Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang diharapkan. Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah-perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian fungsional juga meliputi permukaan yang jelas dari jenis fungsi-fungsi, serta operasi *back-end* (seperti, keamanan dan bagaimana meningkatkan sistem).

2. Pengujian tegangan (*stress testing*)

Pengujian tegangan berkaitan dengan kualitas aplikasi didalam lingkungan. Idenya adalah untuk menciptakan sebuah lingkungan yang lebih menuntut aplikasi, tidak seperti saat aplikasi dijalankan pada beban kerja normal.

3. Pengujian beban (*load testing*)

Pada pengujian beban, aplikasi akan diuji dengan beban berat atau masukan, seperti yang terjadi pada pengujian situs *web*, untuk mengetahui apakah aplikasi atau kinerjanya menurun. Pengujian beban beroperasi pada tingkat beban standar, biasanya beban tertinggi akan diberikan ketika sistem dapat menerima dan tetap berfungsi dengan baik.

4. Pengujian khusus (*ad-hoc testing*)

Jenis pengujian ini dilakukan tanpa penciptaan rencana pengujian (*test plan*) atau kasus pengujian (*test case*). Pengujian khusus membantu dalam menentukan lingkup dan durasi dari berbagai pengujian lainnya dan juga membantu para penguji dalam mempelajari aplikasi sebelum memulai pengujian dengan pengujian lainnya. Pengujian ini merupakan metode pengujian formal yang paling sedikit. Membaca persyaratan atau spesifikasi (jika ada) jarang memberikan panduan yang jelas mengenai bagaimana sebuah program benar-benar bertindak, bahkan dokumentasi pengguna tidak menangkap "*look and feel*" dari sebuah program.

5. Pengujian penyelidikan (*exploratory testing*)

Pengujian penyelidikan mirip dengan pengujian khusus dan dilakukan untuk mempelajari atau mencari aplikasi.

6. Pengujian usabilitas (*usability testing*)

Pengujian ini dilakukan jika antarmuka pengguna dari aplikasinya penting dan harus spesifik untuk jenis pengguna tertentu. Pengujian usabilitas adalah proses yang bekerja dengan pengguna akhir secara langsung maupun tidak langsung untuk menilai bagaimana pengguna merasakan paket perangkat

lunak dan bagaimana mereka berinteraksi dengannya. Tujuan dari pengujian usabilitas harus membatasi dan menghilangkan kesulitan bagi pengguna dan untuk mempengaruhi area yang kuat untuk usabilitas maksimum.

7. “Pengujian Asap” (*smoke testing*)

Pengujian ini dilakukan untuk memeriksa apakah aplikasi tersebut sudah siap untuk pengujian yang lebih besar dan bekerja dengan baik tanpa cela sampai tingkat yang paling diharapkan.

8. Pengujian pemulihan (*recovery testing*)

Pengujian pemulihan (*recovery testing*) pada dasarnya dilakukan untuk memeriksa seberapa cepat dan baiknya aplikasi bisa pulih terhadap semua jenis *crash* atau kegagalan *hardware*, masalah bencana, dan lain-lain.

9. Pengujian *volume* (*volume testing*)

Pengujian *volume* dilakukan terhadap efisiensi dari aplikasi. Pengujian *volume*, seperti namanya, adalah pengujian sebuah sistem (baik perangkat keras dan perangkat lunak) untuk serangkaian pengujian dengan *volume* data yang diproses adalah subjek dari pengujian, seperti sistem yang dapat menangkap sistem pengolahan transaksi penjualan *real-time* atau dapat membarui basis data atau pengembalian data (*data retrieval*).

10. Pengujian domain (*domain testing*)

Pengujian domain merupakan penjelasan yang paling sering menjelaskan teknik pengujian.

11. Pengujian skenario (*scenario testing*)

Pengujian skenario adalah pengujian yang realistis, kredibel dan memotivasi *stakeholders*, tantangan untuk program dan mempermudah penguji untuk

melakukan evaluasi. Pengujian ini menyediakan kombinasi *variable-variable* dan fungsi yang sangat berarti dari pada kombinasi buatan yang didapatkan dengan pengujian domain atau desain pengujian kombinasi.

12. Pengujian regresi (*regression testing*)

Pengujian regresi adalah gaya pengujian yang berfokus pada pengujian ulang (*retesting*) setelah ada perubahan. Pada pengujian regresi berorientasi risiko (*risk-oriented regression testing*), daerah yang sama yang sudah diuji, akan kita uji lagi dengan pengujian yang berbeda (semakin kompleks). Usaha pengujian regresi bertujuan agar mengurangi risiko berikut ini:

- a. Perubahan yang dimaksudkan untuk memperbaiki *bug* yang gagal.
- b. Beberapa perubahan memiliki efek samping, tidak memperbaiki *bug* lama atau memperkenalkan *bug* baru.

13. Penerimaan pengguna (*user acceptance*)

Pada jenis pengujian ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan.

14. Pengujian alfa (*alpha testing*)

Pengguna akan menggunakan aplikasi dan pengembang mencatat setiap masukan atau tindakan yang dilakukan oleh pengguna. Semua jenis perilaku yang tidak normal dari sistem dicatat dan dikoreksi oleh para pengembang.

15. Pengujian beta (*beta testing*)

Pada jenis ini, perangkat lunak didistribusikan sebagai sebuah versi beta dengan pengguna yang menguji aplikasi di situs mereka. Pengecualian atau

cacat yang terjadi akan dilaporkan kepada pengembang. Pengujian beta dilakukan setelah pengujian alfa.