

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Pembahasan tugas akhir ini tidak lepas dari teori-teori yang mendukung kemudahan dalam mempelajari dan merancang program aplikasi.

2.1.1. Website

Konsep dasar *website* berisi tentang definisi *internet*, *website*, basis data, bahasa pemrograman yang digunakan oleh penulis, dan model pengembangan sistem yang digunakan.

A. Internet

Secara umum internet dapat diartikan sebagai media pertukaran informasi dan komunikasi. Semua informasi bisa di dapat dengan mudah dan bebas tanpa ada batasan. Menurut Nugroho (2019:1), "Internet adalah sebuah solusi jaringan dapat menghubungkan beberapa jaringan local yang ada pada suatu Daerah, Kota, atau bahkan pada sebuah Negara. Dengan adanya internet kita dapat menghubungkan beberapa jaringan local yang ada pada setiap tempat".

B. World Wide Web (WWW)

Menurut Sidik & Pohan (2014:1), "*World Wide Web (WWW)*, lebih dikenal dengan web, merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke Internet".

Menurut Purnama (2016:32), "*World Wide Web* adalah kumpulan informasi multimedia (tekstual, grafis, spesial, suara, dan sebagainya) yang disusun dalam bentuk *HyperText*".

C. *Web Browser*

Menurut Purnama (2016:7), “*Web Browser* adalah *software* yang digunakan untuk membuka *web* di *internet*. Jenisnya banyak sekali. Masing-masing mempunyai kelebihan dan kekurangannya. *Web Browser* juga menyesuaikan dengan sistem operasi yang digunakan oleh komputer. Misalnya untuk jalan di *Operating Sistem Windows, Linux* dan bisa berjalan di atas sistem mobile”.

D. *Web Server*

Menurut Yudhanto & Prasetyo, (2018:10), “*Web server* adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui *browser*, kemudian memberikan tanggapan permintaan dalam bentuk halaman situs web atau lebih umumnya dalam dokumen HTML”.

2.1.2. Bahasa Pemrograman

Bahasa Pemrograman yang digunakan dalam perancangan website antara lain :

A. *Hypertext Preprocessor (PHP)*

Menurut Winarno, Zaki & Community (2014:1), “PHP adalah bahasa pemrograman untuk web yang menganut *client server*. Jadi agar bisa berlatih PHP ini, Anda perlu terlebih dahulu menyiapkan *web server* yang bisa menjalankan PHP dan database MYSQL”.

Menurut Mundzir (2018:3), “PHP berasal dari kata “*Hypertext Preprocessor*”, yaitu Bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan bersamaan dengan *HTML*”.

PHP dirancang untuk para pengembang web agar dapat menciptakan suatu halaman *web* yang bersifat dinamis. Artinya PHP dapat membentuk suatu tampilan berdasarkan apa yang diminta oleh *user*, seperti dapat mengakses suatu *database* dan menampilkannya pada halaman *web*.

B. *JavaScript*

Menurut Sidik & Pohan (2014:267), “*Javascript* merupakan modifikasi dari Bahasa c++ dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan asp ataupun internet explorer”.

Menurut Purbadian (2015:54),” *Javascript* merupakan bahasa pemograman script yang berjalan pada sisi client atau browser”.

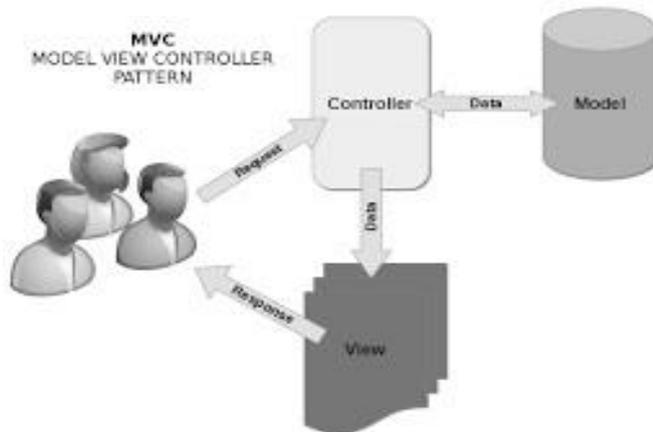
C. *CSS*

Menurut Yudhanto & Prasetyo (2018:4), “*CSS* atau *Cascading Style Sheet* yaitu Bahasa yang digunakan untuk format HTML agar menjadi lebih bagus dan efektif dalam tampilan. Misal, kita ingin membuat paragraph yang memiliki warna merah berulang-ulang maka *CSS* adalah solusi yang tepat”.

Menurut Purbadian (2015:24), “*CSS* yang mempunyai kepanjangan dari *Cascading Style Sheet* merupakan suatu bahasa pemograman web yang digunakan untuk mendesain halaman web (*style sheet language*) agar tercipta suatu halaman web yang rapih, terstruktur, dan seragam”.

D. *MVC*

Menurut Yudhanto & Prasetyo (2018:6), “*Model-View-Controller* atau *MVC* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*)”.



Sumber : Yudhanto & Prasetyo (2018:6)

Gambar II.1 Metode MVC

- a. Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data, seperti memasukkan data ke basis data, pembaruan data, dan lain-lain.
- b. View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
- c. Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

E. Framework

Menurut Sukanto & Shalahuddin (2014:253), “*Framework* adalah kerangka kerja yang memudahkan *programmer* untuk membuat sebuah aplikasi sehingga *programmer* akan lebih mudah melakukan perubahan (*customize*) terhadap aplikasi dan dapat memakainya kembali untuk aplikasi lain yang sejenis”.

Menurut Yudhanto & Prasetyo (2018:8) menyimpulkan bahwa: Framework juga diartikan sebagai kumpulan script (terutama class dan function yang dapat membantu developer/programmer dalam menangani berbagai masalah-masalah dalam pemrograman, seperti koneksi ke *database*,

pemanggilan *variable*, file, dan lain-lain sehingga pekerjaan developer lebih fokus dan lebih cepat dalam membangun aplikasi.

Disini penulis menggunakan *framework Codeigniter* sebagai kerangka kerja pembuatan aplikasi berbasis *web*. Menurut Raharjo (2015:5), “*Codeigniter* adalah *framework* web untuk PHP dan merupakan *framework* yang mengimplementasikan pola desain MVC”.

F. *Jquery*

Menurut Abdulloh (2018:193), “*Jquery* merupakan salah satu dari sekian banyak JavaScript *library*, yaitu kumpulan fungsi JavaScript yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode JavaScript”.

2.1.3. Basis Data

Menurut Yudhanto & Prasetyo (2018:13), “Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi”.

Menurut Yanto (2016:68), “*Database* adalah sekumpulan dari file-file yang dilengkapi dengan atribut dan record yang saling berinteraksi untuk mencapai tujuan menghasilkan informasi yang dibutuhkan oleh pihak pengguna”.

Menurut Sukamto & Shalahuddin (2014:43), “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Berdasarkan pengertian-pengertian diatas, penulis dapat menyimpulkan bahwa basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

A. *XAMPP*

Menurut Sidik (2014:72), “*XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl)* merupakan paket server web *PHP* dan database *MySQL* yang paling populer di kalangan pengembang web dengan menggunakan *PHP* dan *MySQL* sebagai databasenya”.

Paket *XAMPP*, sesuai dengan kepanjangannya, *X* yang berarti *Windows* atau *Linux*, pengguna bisa memilih paket yang diinginkan untuk *Windows* atau *Linux*. *XAMPP* termasuk paket *server* yang paling mudah untuk digunakan sebagai paket untuk pengembangan aplikasi *web*.

B. *MySQL*

MySQL yang merupakan singkatan dari “*My Structured Query Language*” adalah database yang paling favorit saat ini. Menurut Purbadian (2015:2), “*MySQL* adalah sebuah database yang digunakan untuk menampung suatu data”.

Menurut (Supono & Putratama, 2016), “*MySQL* adalah sistem manajemen *database SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database MySQL* mendukung beberapa fitur seperti multithreaded, multiuser, dan *SQL database* manajemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, andal dan mudah digunakan”.

MySQL tidak memiliki aplikasi client berbasis visual seperti database lain (*SQL Server*), sehingga untuk mengakesnya kita memerlukan program aplikasi tambahan. Misalnya *MySQL Front* untuk berbasis desktop, atau *PHPMysqlAdmin* untuk berbasis web.

C. *Structure Query Language (SQL)*

Menurut Sukamto & Shalahuddin (2014:46), “*Structure Query Language (SQL)* adalah bahasa yang digunakan untuk mengelola data pada RDBMS”.

D. Sublime Text

Menurut Supono & Putratama (2016:14) , “Sublime Text merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-*edit* suatu aplikasi. Sublime Text mempunyai fitur *plugin* tambahan yang memudahkan programmer”.

E. PHP My Admin

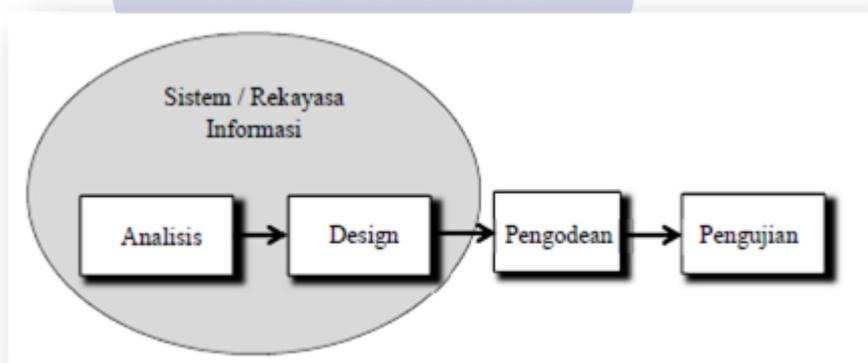
Menurut Yudhanto & Prasetyo (2018:15), “*PHP My Admin* adalah aplikasi web untuk mengelola *database MySQL* dan *database MariaDB* dengan mudah melalui antarmuka (interface) grafis. Aplikasi web ini ditulis menggunakan Bahasa pemrograman *PHP*”.

2.1.4. Model Pengembangan Perangkat Lunak

Menurut Sukamto & Shalahuddin (2014:28) 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 pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).

Berikut adalah gambar model air terjun :



Sumber : Sukamto & Shalahuddin (2014:29)

Gambar II.2 Ilustrasi 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 di dokumentasikan.

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 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 (*support*) atau pemeliharaan (*maintenance*)

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

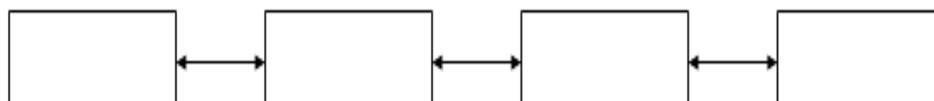
Suatu perancangan program harus didukung dengan suatu peralatan pendukung guna menunjang keberhasilan program dan untuk menjelaskan kepada pengguna bagaimana fungsi dari sistem informasi dapat bekerja dengan suatu logika yang di gambarkan dengan simbol-simbol dan lambang-lambang. *Tools Program* yang penulis gunakan adalah :

2.2.1. Struktur Navigasi

Menurut Andriansyah (2016:61) menjelaskan bahwa, “*Struktur navigasi* adalah dalam sebuah pembuatan *website*, diperlukan *struktur navigasi* sebagai bentuk penggambaran dari halaman-halaman apa saja yang tersedia didalam *website*. Layaknya petunjuk arah, navigasi memudahkan pengguna untuk “berselancar” didalam sebuah *website*.”

1. *Struktur Navigasi Linier*

Pengguna akan melakukan navigasi secara berurutan, dari *frame* atau *byte* informasi ke informasi lainnya.

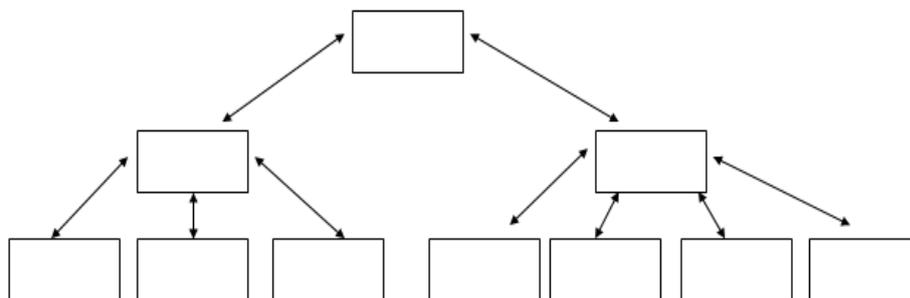


Sumber : Andriansyah (2016:61)

Gambar II.3 Struktur Navigasi *Linier*

2. *Struktur Navigasi Hirarki*

Struktur dasar ini disebut juga struktur”*Linier* dengan percabangan” Karena pengguna melakukan navigasi disepanjang cabang pohon terstruktur yang terbentuk oleh logika isi.

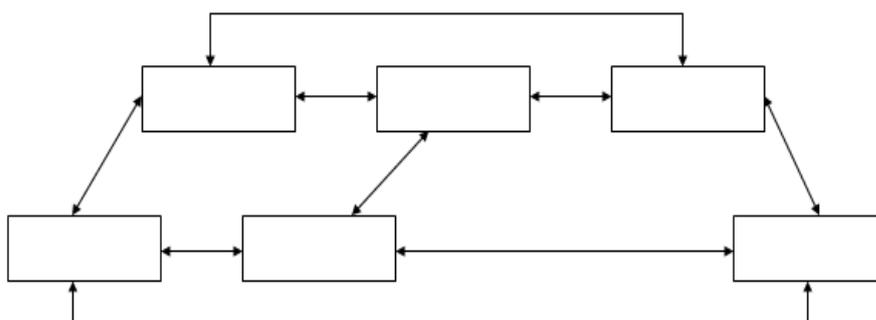


Sumber : Andriansyah (2016:61)

Gambar II.4 Struktur Navigasi *Hirarki*

3. *Struktur Navigasi Non-Linier*

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

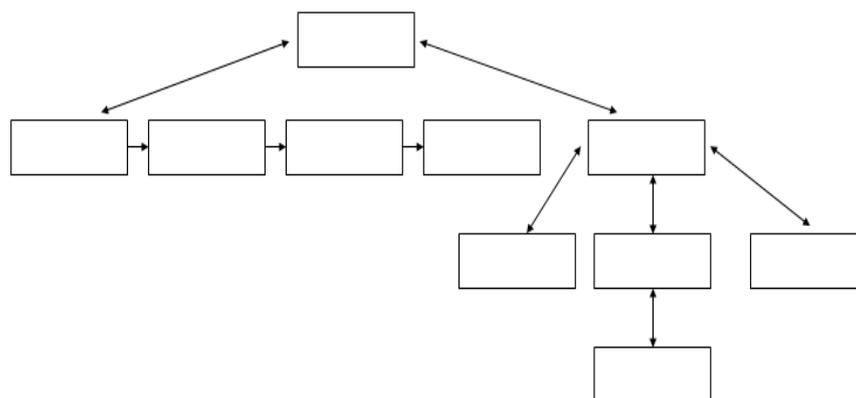


Sumber : Andriansyah (2016:61)

Gambar II.5 Struktur Navigasi *Non-Linier*

4. *Struktur Navigasi campuran (composite)*

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



Sumber : Andriansyah (2016:61)

Gambar II.6 Struktur Navigasi *Composite*

2.2.2. *Entity Relationship Diagram (ERD)*

Menurut Pratama (2014:49), “ERD (Entity Relationship Diagram) adalah diagram yang menggambarkan keterkaitan antartabel beserta dengan field-field di dalamnya pada suatu database sistem”. Sebuah database memuat minimal sebuah tabel dengan sebuah atau beberapa buah *field* (kolom) di dalamnya. Setiap tabel umumnya memiliki keterkaitan hubungan. Keterkaitan antartabel ini biasa disebut dengan relasi.

Menurut Sukamto & Shalahuddin (2015:53), “ERD adalah bentuk suatu model jaringan paling awal dalam melakukan perancangan basis data relasional”. Secara garis besar entitas merupakan dasar yang terlibat dalam sistem. Atribut atau *field* berperan sebagai penjelas dari entitas, dan relasi atau hubungan menunjukkan hubungan yang terjadi antara dua entitas. Berikut akan dijelaskan beberapa simbol yang digunakan dalam ERD Menurut Sukamto & Shalahuddin (2015:50) :

1. Entitas (*Entity*)

Entitas merupakan data inti yang akan disimpan. Penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

2. Atribut

Atribut merupakan *field* atau kolom data yang butuh disimpan dalam suatu entitas.

3. Atribut kunci primer

Merupakan *field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan biasanya berupa id: kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4. Atribut multivalu/*multivalued*

Merupakan *field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5. Relasi

Relasi yang menghubungkan antara entitas yang biasanya diawali dengan kata kerja.

6. Asosiasi atau *association*

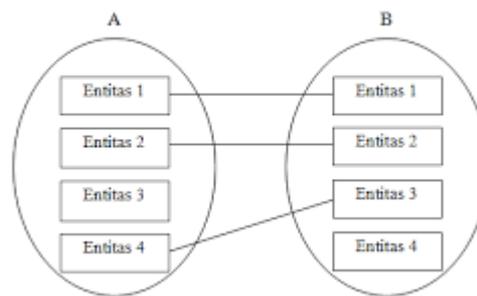
Merupakan penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antar entitas satu dengan entitas yang lain disebut dengan kardinalitas.

A. Derajat Relasi (Kardinalitas)

Menurut Fathansyah (2015:79), kardinalitas relasi yang terjadi di antara dua himpunan entitas (misalnya A dan B) dapat berupa:

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

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas himpunan entitas A.

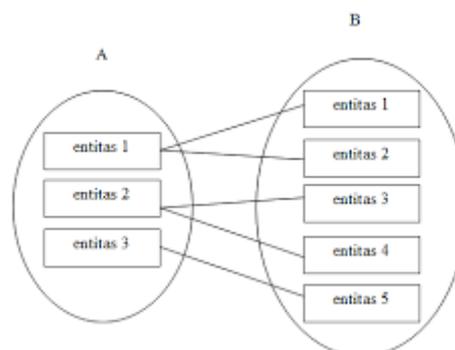


Sumber: Fathansyah (2015:79)

Gambar II.7 Kardinalitas Relasi Satu ke Satu (*One to One*)

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

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

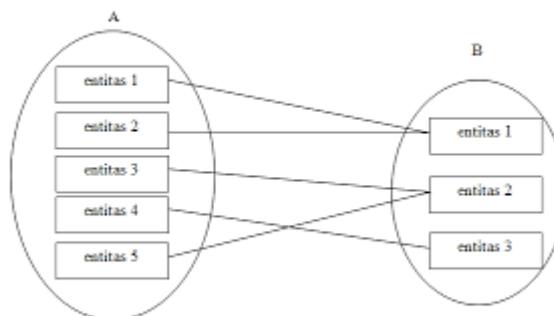


Sumber: Fathansyah (2015:80)

Gambar II.8 Kardinalitas Relasi Satu ke Banyak (*One to Many*)

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

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan banyak dengan satu entitas pada himpunan B, tetapi tidak sebaliknya, setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

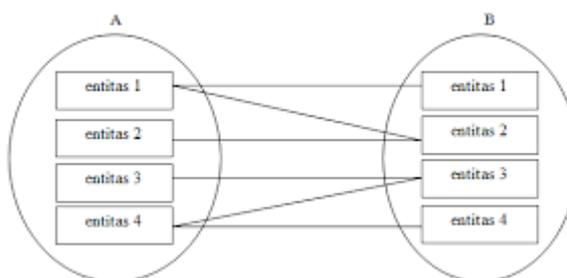


Sumber: Fathansyah (2015:80)

Gambar II.9 Kardinalitas Relasi Banyak Ke Satu (*Many to One*)

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

Yang berarti setiap entitas pada himpunan A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, di mana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



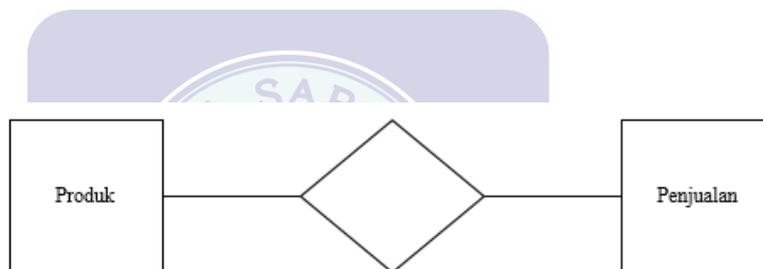
Sumber: Fathansyah (2015:81)

Gambar II.10 Kardinalitas Relasi Banyak ke Banyak (*Many to Many*)

2.2.3. Logical Relational Structure (LRS)

Menurut Andriansyah (2016:53), menjelaskan 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 symbol asterix (*) sebagai symbol primary key (kunci utama) dan foreign key (kunci tamu).

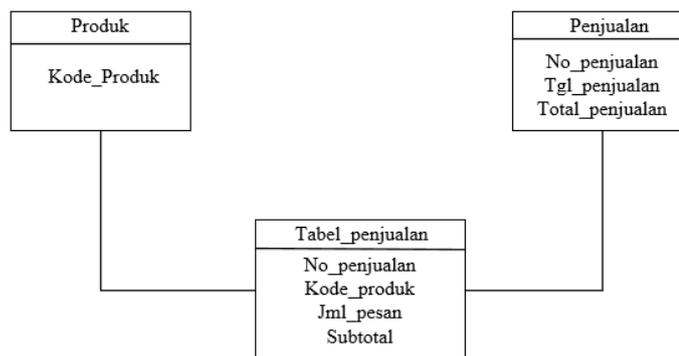
Apabila dalam penggambaran ERD pada point sebelumnya terdapat tabel yang dapat didetail-kan, maka pada LRS dapat digambarkan secara jelas perincian tabel tersebut dengan tabel baru. Sebagai contoh perhatikan gambar di bawah ini:



Sumber: Andriansyah (2016:23)

Gambar II.11 Rancangan ERD Produk dan Penjualan

Pada penggambaran ERD diatas, simbol relasi (belah ketupat) antara tabel produk dan tabel penjualan dapat diisi dengan attribute no_penjualan, kode_produk, jml_pesanan, dan subtotal. Sementara pada tabel penjualan dapat diisi dengan atribut no_penjualan, tgl_penjualan, dan total_penjualan. Transformasi dari ERD diatas kedalam penggambaran LRS adalah sebagai berikut:



Sumber: Andriansyah (2016:23)

Gambar II.12 Rancangan LRS Produk dan Penjualan

2.2.4. Implementasi dan Pengujian Web

Implementasi adalah suatu penerapan dan pelaksanaan kegiatan yang sudah terencana di desain untuk kemudian dijalankan sesuai dengan rancangan.

Menurut Sukamto & Shalahuddin (2014:275) mengemukakan 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”.

Pengujian kotak hitam dilakukan dengan cara 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:

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.

