

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Program

Menurut (Agustini, 2017) “Program adalah kata, pernyataan kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer.”

2.1.1. Website

Menurut (Kirana, 2013) di dalam jurnal (Taufik, 2017) “website atau situs merupakan tempat penyimpanan data dan informasi dengan menggunakan topik tertentu.” Diumpamakan situs web ini adalah sebuah buku yang berisikan sebuah topik tertentu, website atau situs web juga merupakan kumpulan dari halaman-halaman web yang saling berkaitan didalam web tersebut.”

Menurut (Fatmawati, 2016) “*Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses dimana pun selama anda terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi.”

Di dalam sebuah *website*, ada beberapa komponen pendukung agar sebuah *website* dapat diakses oleh pemakai, diantaranya :

1. Internet

Menurut Hidayatullah (2014:1), “*Internet* adalah jaringan *global* yang menghubungkan komputer-komputer diseluruh dunia. Dengan *internet* sebuah komputer bisa mengakses data yang terdapat pada komputer lain di tempat yang berbeda. Dengan *Internet*, beberapa informasi tentang buku yang terdapat di perpustakaan dapat dilihat 24 jam tanpa henti”.

2. Web Browser

Menurut Sidik dan Pohan (2014:5) menjelaskan bahwa “Web Browser adalah software yang digunakan untuk menampilkan informasi web. Software ini kini telah dikembangkan dengan menggunakan *user interface grafis*, sehingga pemakai dapat melakukan *point* dan *click* untuk pindah antar dokumen.

3. Web Server

Menurut Prayitno & Safitri (2015:2) menerangkan bahwa “pengertian *web server* adalah “*server Web (Web Server)* merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protocol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”

2.1.2. Bahasa Pemrograman

Bahasa Pemrograman yang digunakan dalam perancangan website, antara lain:

1. *Hypertext Preprocessor (PHP)*

Menurut (Fatmawati, 2016) “PHP adalah script yang digunakan untuk membuat halaman web yang bersifat dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat suatu halaman itu diminta oleh client. PHP juga bersifat open source sehingga setiap orang dapat menggunakan secara gratis.”

Menurut (Andrea Adelheid, 2012) di dalam jurnal (Taufik, 2017) “PHP *Preprocessor Hypertext* merupakan bahasa berbentuk script yang ditempatkan dalam server dan dieksekusi di dalam server untuk selanjutnya ditransfer dan dibaca oleh client. *PHP* juga bisa disisipkan dalam bahasa *HTML*.”

Menurut Agus Prayitno & Yulia Safitri, (2015:2) menerangkan bahwa “PHP adalah (*PHP Preprocessor Hypertext*) adalah bahasa pemrograman *web* berupa *script* yang dapat diintegrasikan dengan *HTML*”.

2. *JavaScript (JS)*

Menurut Sidik dalam (Agus Prayitno & Yulia Safitri, 2015:2) menjelaskan bahwa “JavaScript adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen *HTML* yang ditampilkan dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja”.

3. *Asynchronous JavaScript and XML (AJAX)*

Menurut Abdullah (2017:2), “AJAX adalah singkatan dari *Asynchronous JavaScript and XML* merupakan solusi untuk permasalahan pada *web* tradisional dimana ketika klien mengirim permintaan ke *server* harus menunggu respon dari server, baru bisa melanjutkan aktivitas berikutnya”. Pada dasarnya *Ajax*

menggunakan *XMLHttpRequest* objek *Javascript* untuk membuat request ke *server* secara *asynchronous* atau tanpa melakukan *refresh* halaman *website*”.

4. *Hyper Text Markup Language (HTML)*

Menurut (Fatmawati, 2016) “*Hypertext Text Markup language* merupakan suatu bahasa yang dikenal oleh *web browser* untuk menampilkan informasi seperti *teks*, gambar, suara, animasi, bahkan video.”

5. *Cascading Style Sheet (CSS)*

Menurut (Fatmawati, 2016) “*CSS (Cascading Style Sheet)* adalah sebuah cara untuk memisahkan isi dengan *layout* dalam halaman-halaman web yang dibuat. *Cascading Style Sheet* dikembangkan untuk menata gaya pengaturan halaman *web*. Pada awalnya *CSS* dikembangkan pada *SGML* pada tahun 1970 dan terus dikembangkan hingga saat ini *CSS* telah mendukung banyak bahasa, *Cascading Style Sheet* memiliki arti gaya menata halaman bertingkat. Yang berarti setiap satu elemen yang telah format, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya.”

6. *JQuery*

Menurut (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*.”

2.1.3. Basis Data

1. Basis Data (*Database*)

Menurut Sukanto dan Shalahuddin (2015:42) menerangkan bahwa “Basis Data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah di olah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

2. *MySQL*

a. Pengertian *MySQL*

Menurut (Agustini, 2017) “*MySQL* adalah *multiuser database* yang menggunakan bahasa *structured query language (SQL)*. *MySQL* merupakan *software* yang tergolong sebagai *DBMS (Database Management System)* yang bersifat *open source*”.

Menurut (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*.”

b. Memahami Perintah *MySQL*

Adapun perintah yang terdapat pada *MySQL* adalah sebagai berikut :

Tabel II.1. Perintah Dasar *MySQL*

Perintah	Keterangan
<i>Show databases</i>	Perintah ini digunakan untuk menampilkan atau melihat daftar <i>database</i> yang sudah ada (sudah dibuat).
<i>Use</i>	Perintah ini digunakan untuk masuk atau mengakses <i>database</i> yang sudah ada.
<i>Show tables</i>	Perintah ini digunakan untuk melihat atau menampilkan semua tabel yang ada di dalam <i>database</i> aktif (yang sudah dibuka, sudah di <i>use</i>).
<i>Desc/describe</i>	Perintah ini digunakan untuk melihat struktur tabel.
<i>Quit</i>	Perintah ini digunakan untuk keluar <i>MySQL Server</i> .

3. XAMPP

a. Pengertian XAMPP

Menurut (Fatmawati, 2016) “*Xampp* adalah salah satu paket *software web server* yang terdiri dari *Apache*, *Mysql*, *Php* dan *PhpMyAdmin*. Proses instalasi *xampp* sangat mudah, karena tidak perlu memerlukan konfigurasi *Apache*, *Php*, dan *Mysql* secara manual, *xampp* melakukan instalasi dan konfigurasi secara otomatis.”

Menurut Nugroho (2014:1), “*XAMPP* adalah paket program *web* lengkap yang dapat anda pakai untuk belajar pemrograman *web*, khususnya *PHP* dan *MySQL*, paket ini dapat didownload secara gratis dan legal”.

Sedangkan menurut Sandi (2014:31), “*XAMPP* adalah distribusi *Apache* kecil dan ringan yang mengandung teknologi pengembangan *web* yang paling umum dalam satu paket. Isinya, ukuran kecil, dan portabilitas membuat alat yang ideal bagi pelajar untuk mengembangkan dan menguji aplikasi di *PHP* dan *MySQL*”.

b. Memahami Folder XAMPP

Menurut Nugroho (2014:6), di dalam folder utama *xampp*, terdapat beberapa *folder* penting yang perlu diketahui. Untuk lebih memahami setiap fungsinya, Anda dapat melihat penjelasannya sebagai berikut :

Tabel II.2. Folder Penting XAMPP

Folder	Keterangan
<i>Apache</i>	<i>Folder</i> utama dari <i>Apache Webserver</i> .
<i>Htdocs</i>	<i>Folder</i> utama untuk menyimpan data-data latihan <i>web</i> , baik <i>PHP</i> maupun <i>HTML</i> biasa.

<i>Manual</i>	Berisi subfolder yang di dalamnya terdapat manual program dan <i>database</i> , termasuk manual <i>PHP</i> dan <i>MySQL</i> .
<i>MySQL</i>	Folder utama untuk database <i>MySQLServer</i> .
<i>PHP</i>	Folder utama untuk program <i>PHP</i> .

(Sumber : Nugroho (2014:7)).

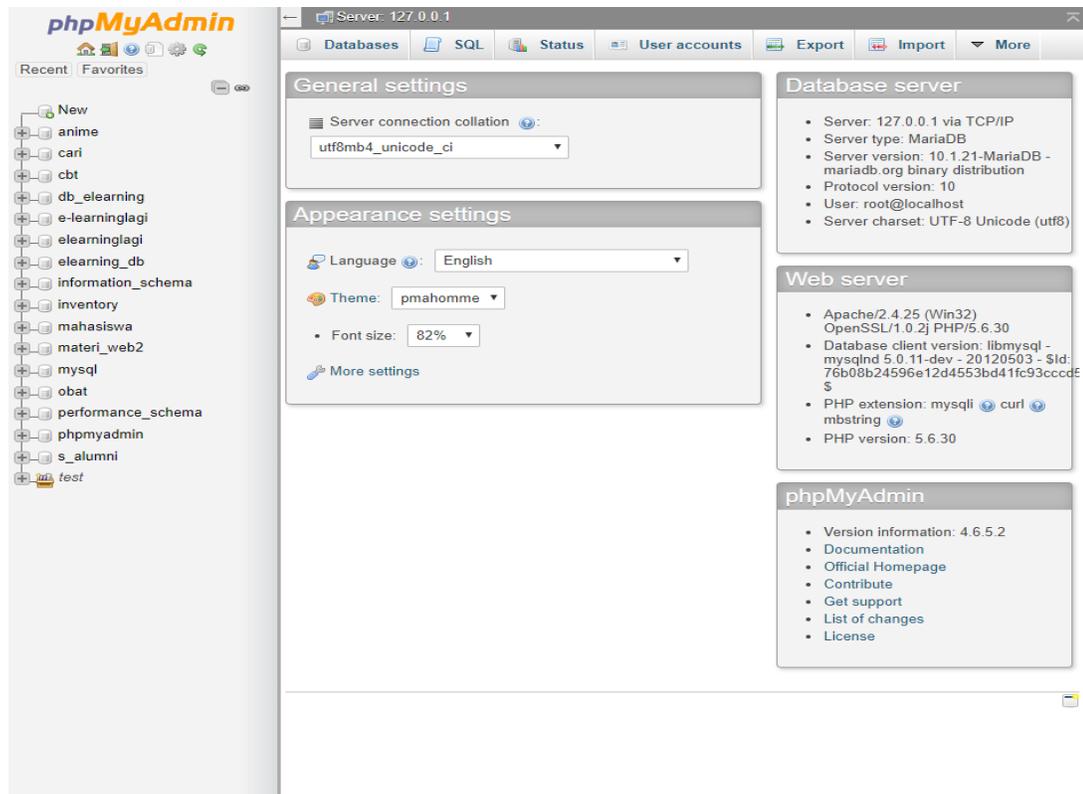
4. PHP MyAdmin

a. Pengertian php MyAdmin

Menurut (Barri, 2015) “PhpMyadmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (World Wide Web). phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel- tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perijinan (permissions), dan lain-lain). Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (command line) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (database), ketikkan baris perintah yang sesuai untuk membuat basis data.

Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu. Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat

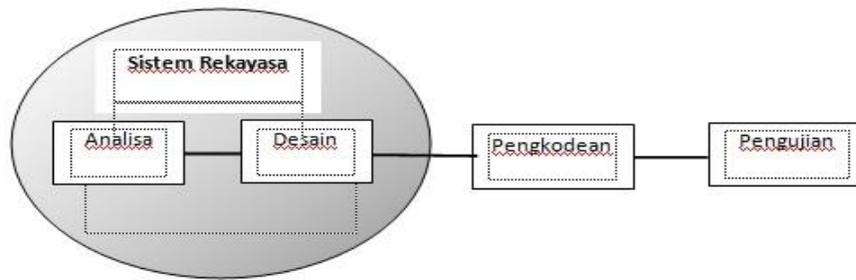
tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya. Php MyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer (www. Wikipedia. org.2011)”. Berikut tampilan dari *phpMyAdmin* :



Gambar II.1. Tampilan dari *phpMyAdmin*

2.1.4. Model *Waterfall*

Menurut Sukamto dan Shalahuddin (2014:28) Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) 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*). Berikut adalah gambar model air terjun :



Sumber : Sukamto dan Shalahuddin (2014:28)

Gambar II.2. Ilustrasi *model waterfall*

2.2. Teori Pendukung

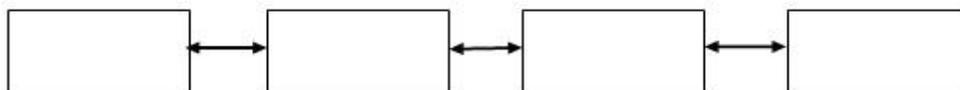
2.2.1. Struktur Navigasi

Menurut Andriansyah (2016:61) “Struktur Navigasi dapat diartikan sebagai alur dari suatu *program* yang menggambarkan rancangan hubungan antar *area* yang berbeda sehingga memudahkan proses pengorganisasian seluruh *element website*”..

Bentuk dasar dari struktur navigasi sebagai berikut :

1. Struktur Navigasi Berurut (*Linear*)

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.

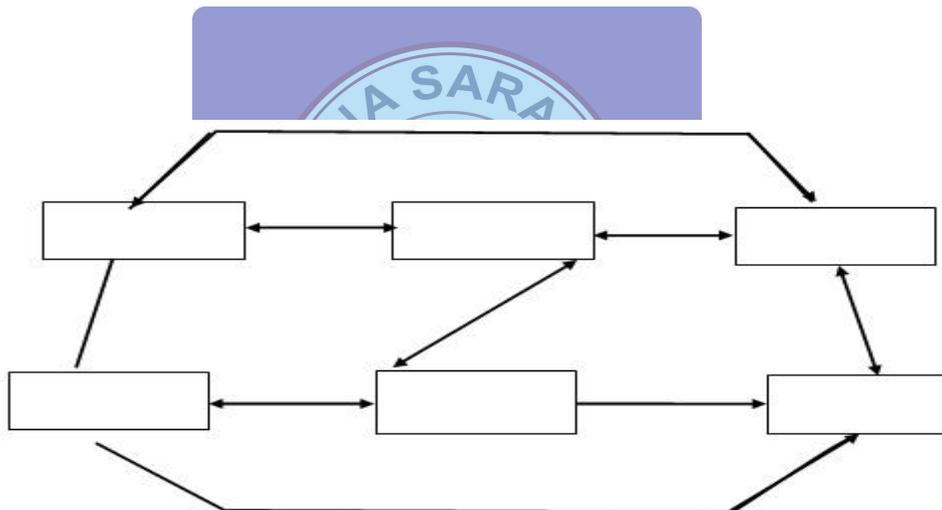


Sumber : Andriansyah, (2016:62)

Gambar II.3 Struktur Navigasi Berurut (*Linear*)

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

Struktur navigasi *non-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 ada percabangan *non-linier* ini walaupun terdapat percabangan tetep 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.

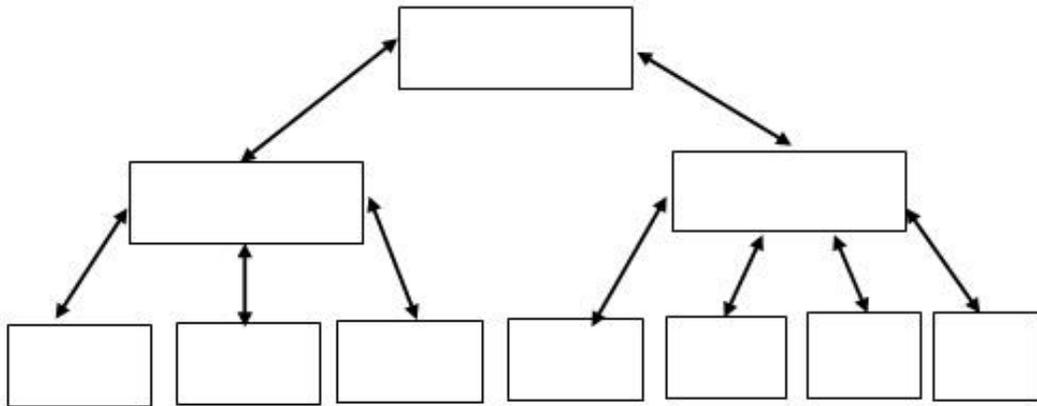


Sumber : Andriansyah (2016:62)

Gambar II.5 Struktur Navigasi Tidak Berurut (*Non-Linear*)

3. Struktur Navigasi Hierarki (*Hierarchi*)

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

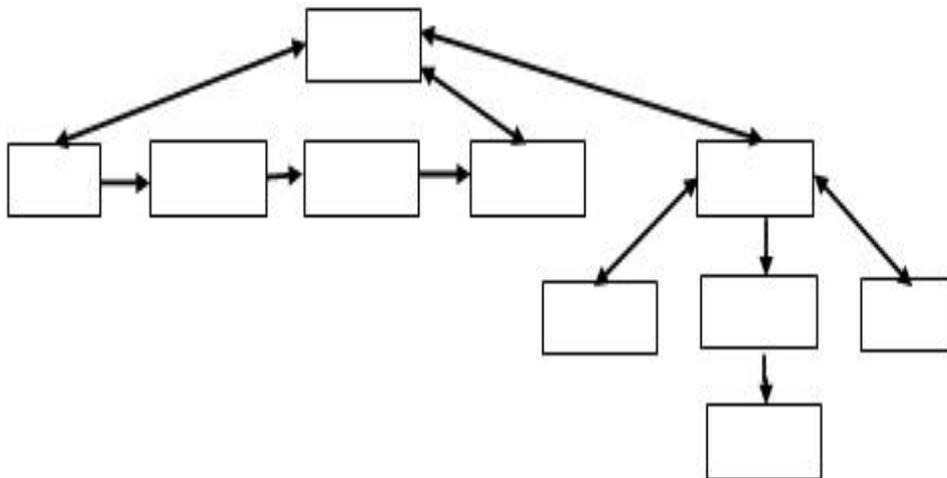


Sumber : Andriansyah (2016:62)

Gambar II.4 Struktur Navigasi Hierarki (*Hierarchi*)

4. Struktur Navigasi Campuran (*Composite*)

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



Sumber : Andriansyah (2016:63)

Gambar II.6 Struktur Navigasi (*Composite*)

2.2.2. Entity Relational Diagram (ERD)

Menurut (Fatta, 2007) di dalam jurnal (Taufik, 2017) “*Entity Relationship Diagram (ERD)* adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis.”

1. Entity

Menurut (Indrajani, 2011) di dalam jurnal (Taufik, 2017) “*Entity* yaitu kumpulan objek-objek dengan sifat (property) yang sama, yang diidentifikasi oleh *enterprise* mempunyai eksistensi yang *independen*” Entitas diberi nama dengan kata benda dan dapat dikelompokkan sebagai berikut:

- a. Nama Orang
- b. Nama benda
- c. Nama lokasi
- d. Nama Kejadian (terdapat unsur waktu didalamnya)

2. Relationship

Menurut (Indrajani, 2011) di dalam jurnal (Taufik, 2017) relationship adalah kumpulan keterhubungan yang mempunyai arti antara tipe entitas yang ada. Pada umumnya penghubung diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya. Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang

3. Relationship Degree

Menurut (Indrajani, 2011) di dalam jurnal (Taufik, 2017) “*Relationship Degree* atau *derajat relationship* adalah jumlah entitas yang berpartisipasi dalam satu



relationship.” *Derajat Relationship* yang sering dipakai di dalam *ERD* sebagai berikut:

- a. *Unary Relationship* *Unary Relationship* adalah *model relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama.
- b. *Binary Relationship* *Binary Relationship* adalah *model relationship* antar *instance-instance* untuk suatu tipe entitas.
- c. *Ternary Relationship* *Ternary Relationship* adalah merupakan *relationship instance-instance* dari tiga tipe entitas secara sepihak.

4. Atribut

Menurut (Indrajani, 2011) di dalam jurnal (Taufik, 2017) “Atribut merupakan sifat-sifat (*property*) sebuah *entity* atau *tipe relationship*.”

Attribute value adalah suatu *occurrence* tertentu dari sebuah *attribute* didalam suatu *entity* atau *relationship*. Ada dua jenis atribut:

- a. *Identifier (key)* digunakan untuk menentukan sesuatu *entity* secara unik (*primary key*).
- b. *Descriptor (nonkey attribute)* digunakan untuk menspesifikasikan karakteristik dari suatu *entity* yang tidak unik.

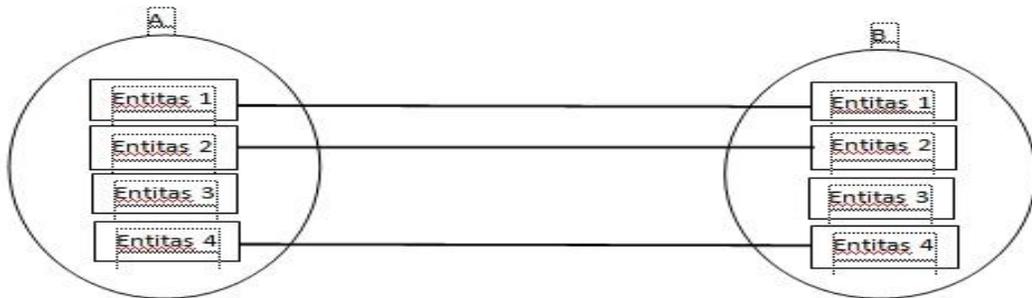
Menurut Pratama (2014:49) “*Entity Relational Diagram*(*ERD*) adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan field-field didalamnya pada suatu database sistem”. Sebuah database membuat minimal sebuah tabel dengan sebuah *field* atau beberapa buah *field* (kolom). Namun pada kenyataanya *database* sering memiliki lebih dari satu buah tabel (dengan beberapa *field* didalamnya). Setiap

tabel umumnya memiliki keterkaitan hubungan. Keterkaitan antar tabel ini disebut dengan relasi. Terdapat 4 buah jenis relasi antar tabel di dalam bagan.

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

Berarti setiap entitas pada himpunan entitas A berhubungan dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan B berhubungan dengan satu entitas B

Contoh :



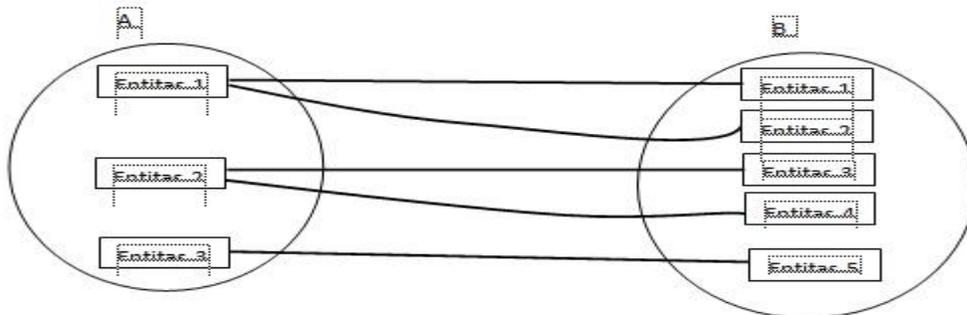
Sumber: Pratama (2014:49)

Gambar II.7 Relasi *One to One*

2. *One to many* (Satu ke banyak)

Entity pertama dapat berhubungan dengan sejumlah *entity* kedua, tetapi satu *entity* kedua hanya dapat berhubungan dengan satu *entity* kedua.

Contoh :



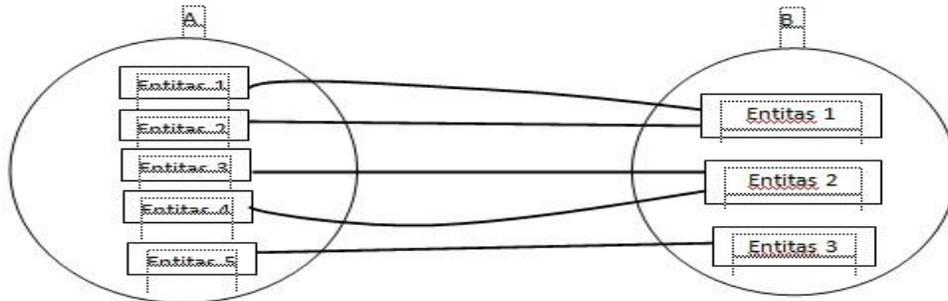
Sumber: Pratama (2014:49)

Gambar II.8 Relasi *One to Many*

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

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, tetapi sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

Contoh :

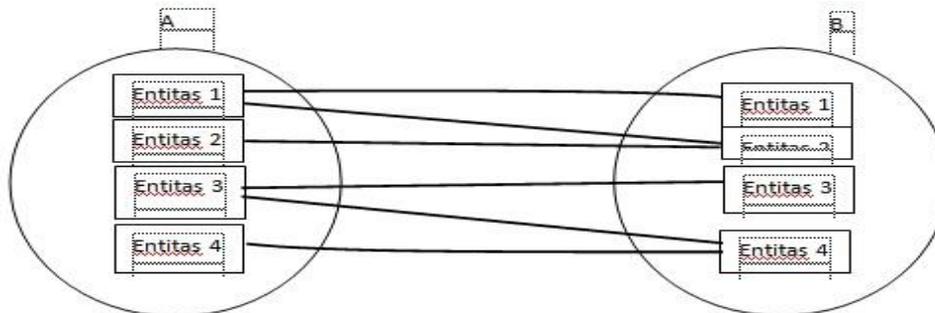


Sumber: Pratama (2014:49)

Gambar II.9 Relasi *Many to One*

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

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya, di mana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Sumber: Pratama (2014:49)

Gambar II.10 Relasi *Many to many*

2.2.3. Logical Record Structure (LRS)

Menurut (Friyadie, 2007) di dalam jurnal (Taufik, 2017) “sebelum tabel dibentuk dari *field* atau *atribut* entitas secara fisik atau *level internal*, maka harus dibuatkan suatu bentuk *relational model* yang dibuat secara *logic* atau *level external* dan konsep, dari pernyataan tersebut dibutuhkan yang disebut dengan *Logical Record Structure (LRS)*”.

Dalam pembuatan *Logical Record Structure (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 (*Black-Box testing*)

Menurut (Ilmi, 2017) mengatakan “Pengujian dan implementasi sistem yang dilakukan dengan menggunakan metode *black box testing*”, berikut hal-hal yang diuji diantaranya:

- a. Fungsi yang hilang atau tidak benar

Fungsi dapat berjalan dengan benar sesuai dengan proses yang ditentukan sehingga semua dapat berfungsi sesuai prosedur

b. Kesalahan *Interface*

Kesalahan *interface* program harus dipastikan dapat berjalan dengan benar dan program yang dibuat sesuai dengan rancangan dan kebutuhan

c. Kesalahan dalam struktur data atau akses *database eksternal*.

