

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Rujukan penelitian yang pertama yaitu jurnal Dedy Hartama dan Muhammad Zarlis mahasiswa program *magister* (S2) Universitas Sumatera Utara pada tahun 2010 dengan judul Aplikasi E-Tugas Berbasis Web CMS Studi Kasus Amik Tunas Bangsa Pematangsiantar. Dalam penelitiannya peneliti menggunakan aplikasi yang menggunakan bahasa pemrograman PHP untuk proses *upload* tugas mahasiswa dan memberikan nilai pada mahasiswa. Dimana mahasiswa mengisi form melalui aplikasi untuk melakukan *upload* tugas kuliah, dan dosen mengisi form untuk memberikan nilai tugas kepada mahasiswa.

Rujukan penelitian yang kedua yaitu jurnal Arman Suryadi Karim mahasiswa Institut Informatika dan Bisnis Darmajaya pada tahun 2016 dengan judul Sistem Informasi Tugas Akhir Berbasis Web. Dalam penelitiannya peneliti menggunakan aplikasi yang menggunakan Microsoft.NET 3.0 *framework* sebagai penyedia komponen yang diperlukan untuk membangun sistem dan objek, ditambah dengan penyediaan sistem akses data yang dibutuhkan oleh komponen sistem. Sistem ini diimplementasikan dengan menggunakan bahasa pemrograman ASP. NET sebagai alat untuk desain halaman web yang dikombinasikan dengan VB. NET untuk proses *upload* pengajuan judul tugas akhir mahasiswa. Dimana mahasiswa mengisi form melalui aplikasi untuk melakukan proses *upload* pengajuan judul tugas akhir mahasiswa.

Sedangkan penelitian yang akan dilakukan oleh peneliti tidak jauh berbeda dengan penelitian sebelumnya yaitu sebagai tempat *upload* dan *download* dalam pengumpulan tugas kuliah mahasiswa di Jurusan Teknik Komputer Polstri berbasis *website*. Dengan menggunakan aplikasi yang menggunakan bahasa pemrograman PHP. Dimana dosen akan *upload* tugas melalui aplikasi untuk memberikan tugas pada mahasiswa, dan mahasiswa bisa mengunduh tugas yang diberikah oleh dosen serta bisa *upload* tugas, Selain itu di aplikasi ini dosen akan dimudahkan lagi dalam

pendataan tugas kuliah mahasiswa akan adanya notifikasi pada *WhatsApp* dosen terkait.

Untuk lebih jelas dan detail terhadap penelitian terdahulu dapat dilihat pada tabel berikut:

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian Sekarang

No.	Penelitian	Persamaan	Perbedaan
1.	Hartama, Dedy, dan Muhammad Zarlis. 2010. Aplikasi E-Tugas Berbasis Web CMS Studi Kasus Amik Tunas Bangsa Pematangsiantar.	<ul style="list-style-type: none"> - Database Menggunakan MySQL. - Menggunakan bahasa pemrograman PHP untuk proses <i>upload</i> tugas mahasiswa. 	<ul style="list-style-type: none"> - Pengaplikasian digunakan hanya pada jurusan Teknik Komputer POLSRI. - Notifikasi pada Dosen terkait melalui aplikasi <i>WhatsApp</i> untuk memudahkan Dosen dalam pendataan dan pengecekan tugas kuliah mahasiswa.
2.	Karim, Arman Suryadi. 2016. Sistem Informasi Tugas Akhir Berbasis Web.	<ul style="list-style-type: none"> - Dalam aplikasi ini mahasiswa mengisi form melalui aplikasi untuk melakukan proses <i>upload</i> tugas mahasiswa. 	<ul style="list-style-type: none"> - Database Menggunakan MySQL. - Menggunakan bahasa pemrograman PHP untuk proses <i>upload</i> tugas mahasiswa.

2.2 Aplikasi

Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jadi aplikasi secara umum adalah suatu proses dari

cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal (Jogiyanto, 2014).

Aplikasi adalah *software* atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus (Daryanto, 2014).

Aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word dan Microsoft Excel (Asropudin, 2013).

Dari uraian diatas dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data serta bertujuan membantu manusia dalam mengerjakan tugas tertentu.

2.3 Pengertian Web

Web adalah layanan sajian informasi yang memudahkan *surfer* (sebutan para pemakai komputer) yang melakukan *browsing*, terdiri dari perangkat lunak, kumpulan protokol, dan seperangkat aturan (Ramadhan, 2015).

2.4 Pengertian PHP

PHP (*Hypertext Processor*) disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap (Andre, 2015).



Gambar 2.1 Tampilan Logo PHP

(Sumber : www.duniaikom.com)

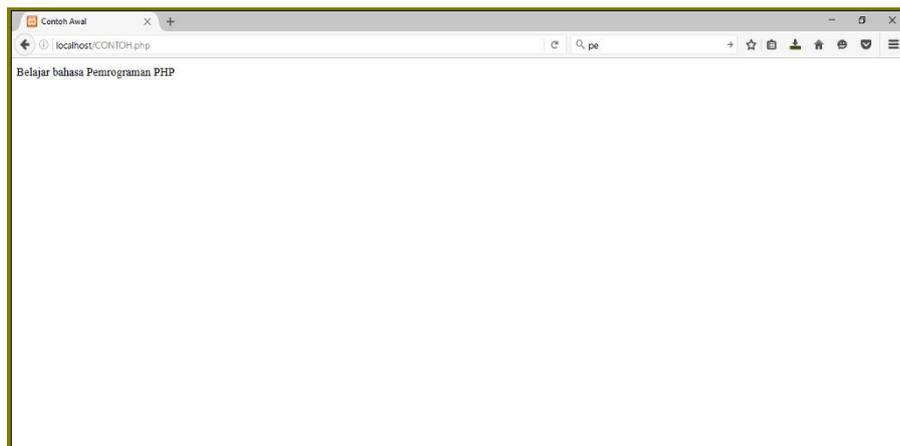
2.4.1 Sintaks Dasar PHP

Kode (*script*) PHP yang sering disebut dengan istilah embedded script yaitu script PHP yang disisipkan di antara script HTML. Jadi dapat dikatakan script PHP hanya ditulis atau disisipkan ketika dibutuhkan saja, seperti menampilkan data dari database meng-upload file, delete data, edit data dan lain sebagainya (Ramadan, 2015).

Contoh *script* :

```
<HTML>
  <HEAD>
    <TITLE>Contoh Awal</TITLE>
  </HEAD>
  <BODY>

    <?php
      echo "Belajar bahasa Pemrograman PHP";
    ?>
  </BODY>
</HTML>
```



Gambar 2.2 Contoh *Script* PHP

(Sumber : Ramadan, 2015).

2.4.2 Tipe Data PHP

Tipe data PHP digunakan untuk menentukan jenis data yang akan disimpan dalam suatu variabel (Ramadhan, 2015). PHP mempunyai empat tipe data dasar sebagai berikut :

1. **Integer** merupakan tipe data yang mencakup semua bilangan bulat. *Range* bilangan integer adalah antara -2.147.4833.647 sampai dengan 2.147.483.647
2. **Floating point** merupakan tipe data yang mencakup semua bilangan desimal (bilangan yang memiliki angka dibelakang koma). *Range* bilangan floating point antara 1e308 sampai dengan 1e308.
3. **Character** merupakan tipe yang digunakan untuk menyimpan data-data yang berupa karakter (satu huruf). Penulisannya biasanya diapit dengan tanda kutip satu ('...').
4. **String** merupakan tipe data tersendiri dan tidak dapat dikelompokkan menjadi tipe data dasar. Penulisannya biasa diapit dengan tanda kutip dua ("...").

2.5 Pengertian Database

Database adalah kumpulan informasi dan data yang saling berhubungan satu sama lain, dimana data tersebut tersimpan di simpanan luar komputer dan diperlukan software tertentu untuk memanipulasinya (Jogiyanto, 2015).

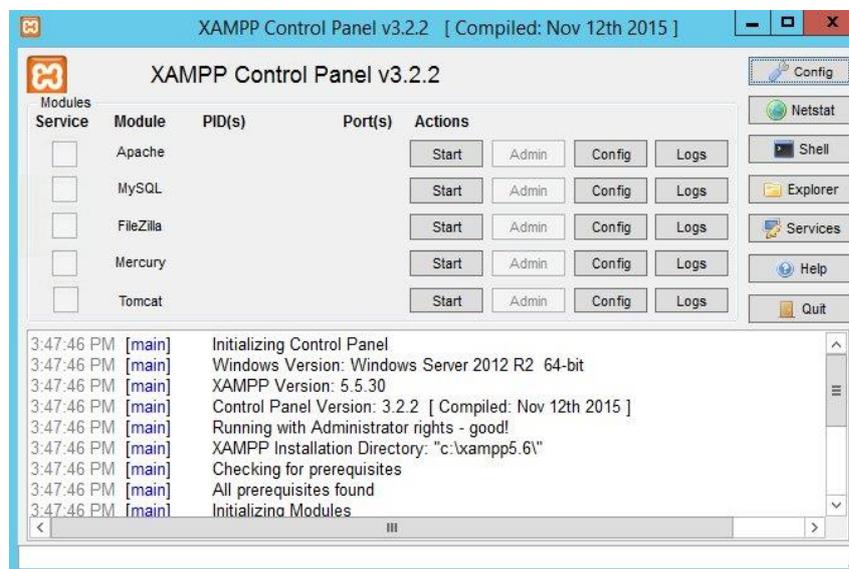
Database atau basis data adalah suatu pengorganisasian sekumpulan data yang saling terhubung sehingga memudahkan aktivitas untuk memperoleh informasi (Kadir, 2013).

Dari uraian diatas dapat disimpulkan bahwa *database* adalah sekumpulan informasi serta data yang saling terhubung, dimana data tersebut disimpan melalui *software database* tertentu untuk memanipulasi data nya. Adapun peneliti menggunakan database MySQL (*My Structure Query Language*) yang dijalankan di *software* XAMPP.

2.6 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer local. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *Panel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

Software XAMPP dapat diperoleh secara gratis dari situs www.apachefriends.org atau dapat juga dicari di *google* untuk beberapa versi tertentu. XAMPP adalah perangkat lunak yang bebas, yang mendukung banyak sistem operasi dan merupakan komplikasi dari beberapa program. Fungsi adalah sebagai *server* yang berdiri sendiri yang terdiri dari atas program *apache* HTTP *server*, *MySQL database*, dan penerjemah bahasa yang ditulis dalam dengan pemrograman PHP dan Perl (Kadir, 2015).



Gambar 2.3 Tampilan Awal XAMPP

(Sumber :Kadir, 2015).

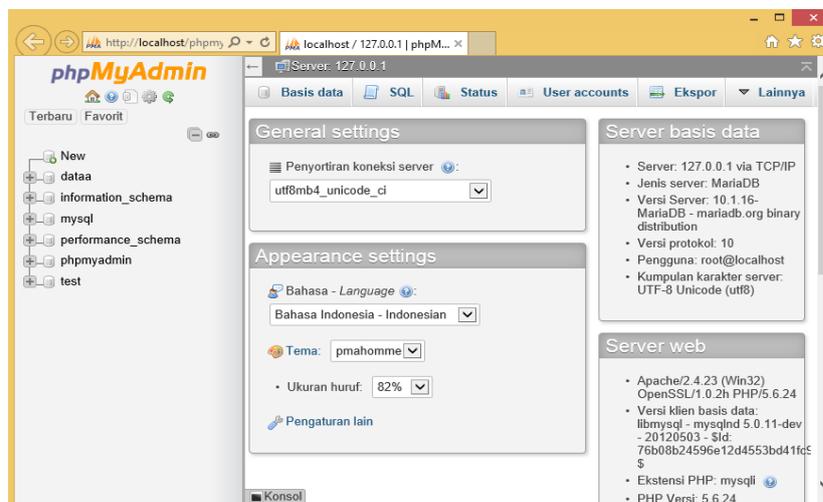
2.7 Pengertian MySQL

MySQL (*My Structure Query Language*) adalah sebuah program pembuat database yang bersifat open source. MySQL sebenarnya produk yang berjalan pada platform Linux. Karena sifatnya *open source*, maka MySQL dapat dijalankan pada semua platform baik Windows maupun Linux. Selain itu, MySQL juga merupakan program pengakses database yang bersifat jaringan sehingga dapat digunakan untuk aplikasi Multi User (Banyak Pengguna). Saat ini database MySQL telah digunakan hampir oleh semua *programer database*, apalagi dalam pemrograman web.

Kelebihan lain dari MySQL adalah menggunakan bahasa *query* standar yang dimiliki SQL (*Structure Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur yang telah di standarkan untuk semua program pengakses database seperti Oracle, Posgres SQL, SQL Server, dan lain-lain (Kadir, 2015).

2.8 PhpMyAdmin

PhpMyAdmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan PhpMyadmin, dapat memudahkan dalam membuat database, membuat tabel, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual (Kadir, 2015).



Gambar 2.4 Tampilan PhpMyAdmin
(Sumber :Kadir, 2015).

2.9 Pengertian CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, subbab, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

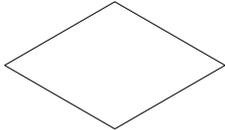
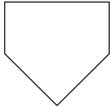
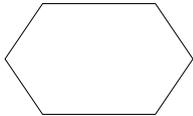
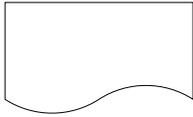
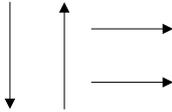
CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda (Azzam, 2014).

2.10 Pengertian Flowchart

Flowchart adalah suatu bagan yang menggambarkan arus logika dari data yang akan di proses dalam suatu program dari awal sampai akhir. Bagan alir program yang berguna bagi *programmer* untuk mempersiapkan program yang lebih kompleks. Bagan alir terdiri dari simbol yang mewakili fungsi-fungsi langkah program dan garis alir yang menunjukkan urutan dari simbol yang akan dikerjakan (Hartono, 2015).

Tabel 2.2 Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
	<i>Input/output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya.

	<i>Process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
	<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
	<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	<i>External Card</i>	Menyatakan input berasal dari kartu atau output data ke kartu.
	<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui Printer)
	<i>Flow</i>	Menyatakan jalannya arus suatu proses

(Sumber: Hartono, 2015).

2.11 Pengertian Data Flow Diagram (DFD)

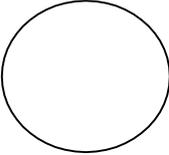
Data Flow Diagram merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data

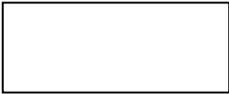
tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Kristanto, 2014).

Data Flow Diagram atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (input) dan keluaran (output). DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek (Sukanto, 2014).

Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.3 Simbol-simbol Data *Flow* Diagram (DFD)

NOTASI	KETERANGAN
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p>
	<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>) pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CMD)</i>, <i>Physical Data Model (PDM)</i>).</p>

	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p>
	<p>Aliran data merupakan data yang di kirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p>

(Sumber :Sukamto, 2014).

Sukamoto (2014:72) Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan Data *Flow* Diagram (DFD) :

1. **Data Flow Diagram (DFD) Level 0**

DFD *Level 0* menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD *Level 0* digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. **Data Flow Diagram (DFD) Level 1**

DFD *Level 1* digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD *Level 1* merupakan hasil *breakdown* DFD *Level 0* yang sebelumnya sudah dibuat.

3. **Data Flow Diagram (DFD) Level 2**

Modul-modul pada DFD *Level 1* dapat di *breakdown* menjadi DFD *Level 2*. Modul mana saja yang harus di *breakdown* lebih detail tergantung pada tingkat kedetilan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di *breakdown* lagi. Untuk sebuah sistem, jumlah DFD *Level 2* sama dengan jumlah modul pada DFD *Level 1* yang di *breakdown*.

4. Data Flow Diagram (DFD) Level 3 dan seterusnya.

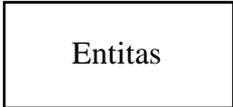
DFD *Level 3*, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD *Level* di atasnya. *Breakdown* pada *level 3*, 4 dan 5 dan seterusnya aturannya sama persis dengan DFD *Level 1* atau *Level 2*.

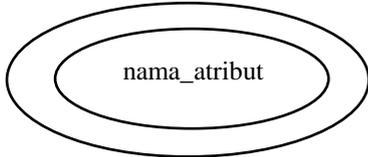
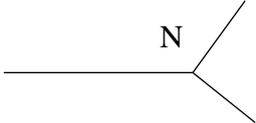
2.12 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional. 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. Namun yang banyak digunakan adalah notasi *Chen* (Sukamto, 2014).

Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen :

Tabel 2.4 Simbol-simbol *Entity Relationship Diagram* (ERD)

SIMBOL	DESKRIPSI
	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> 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).
<p>Atribut multinilai/<i>multivalued</i></p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
<p>Asosiasi/<i>association</i></p> 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian, keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

(Sumber :Sukamto, 2014).