

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

(Indrajani, 2015), dalam penelitian yang berjudul “Perancangan Sistem Basis Data pada Klinik”, merupakan penelitian yang bertujuan untuk menganalisis dan merancang basis data untuk mendukung kebutuhan informasi klinik khususnya pada pendaftaran pasien dan pengobatan pasien. Metode yang digunakan adalah metode pengumpulan data menggunakan teknik pencarian data yang meliputi wawancara, mempelajari dokumen, observasi, dan studi kepustakaan.

Perancangan *database* berbasis *PHPMyAdmin* juga pernah dilakukan sebelumnya dalam skripsi yang berjudul “Perancangan *Database* Berbasis *PHPMyAdmin* Pada Situs Pariwisata Pendukung Proses Pembuatan Rencana Program *Tourism*”. Skripsi tersebut bertujuan untuk Merancang dan mengembangkan *database* pada *website* perencanaan pariwisata untuk membantu turis dalam membuat rencana pariwisata sendiri secara detail dan jelas. Metode yang digunakan dalam perancangan *database* tersebut meliputi 4 tahapan yaitu *Requirement Collection and Analysis* (Pengumpulan dan Analisa Data), Perancangan *Database* Konseptual (Perancangan *Entity Relationship Diagram*), Perancangan *Database* Logikal (Perancangan *Relational Database*), Perancangan *Database* Fisikal (Struktur-struktur penyimpanan pada *file-file database*). (Ariyoga 2016)

(Abdillah, 2006), dalam penelitian yang berjudul “Perancangan Basisdata Sistem Informasi Penggajian (Studi Kasus pada Universitas (‘XYZ’))”, merupakan perancangan *database* yang berfungsi untuk membantu bagian pengembangan perangkat lunak dalam membuat Sistem Informasi Penggajian. Tujuan dilakukannya penelitian ini adalah untuk membuat skema perancangan *database* (basisdata) untuk Sistem Informasi Penggajian pada Universitas ‘XYZ’. Metode perancangan basisdata yang digunakan adalah *conceptual scheme design* yang merupakan perancangan *Entity Relationship Diagram* (ERD), *logical design* yang

merupakan perancangan *Relational database* dan *phisycal design* yang merupakan perancangan struktur penyimpanan pada *file-dile database*.

Berdasarkan pada tiga penelitian di atas dapat diambil kesimpulan bahwa terdapat beberapa tahapan dalam melakukan perancangan *database* dalam pembuatan *website* yaitu pengumpulan dan analisa data dengan teknik pencarian data yang meliputi wawancara, mempelajari dokumen, observasi, dan studi kepustakaan, perancangan basis data konseptual, perancangan basis data logikal dan perancangan basis data fisik. Sedangkan dalam penelitian ini penulis akan melakukan perancangan *database* untuk pembuatan aplikasi *android* dengan metode perancangan *database* yang meliputi Analisa dan Pengumpulan Data, Perancangan *Database Level* Konseptual, Perancangan *Database Level* Logik, dan Perancangan *Database Level* Fisik.

2.2 Landasan Teori

2.2.1 Pengertian *Database*

(Connoly, 2010) *database* adalah kumpulan data dan deskripsi data yang terhubung secara logika serta dirancang untuk memenuhi kebutuhan informasi untuk suatu organisasi.

(Sutarman, 2012) *database* adalah sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya.

(Kristanto.Ir., 1994) *Database* adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain. Sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. Bila terdapat *file* yang tidak dapat dipadukan atau dihubungkan dengan *file* yang lainnya berarti *file* tersebut bukanlah kelompok dari satu *database*, *file* tersebut dapat membentuk satu *database* sendiri.

2.2.2 Perancangan *Database*

(Connolly, 2002, p. 418), Perancangan *database* merupakan proses menciptakan perancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan. Dalam merancang suatu basis data, digunakan metodologi-metodologi yang membantu dalam tahap perancangan basis data. Metodologi perancangan adalah pendekatan struktur dengan menggunakan prosedur, teknik, alat, serta bantuan dokumen untuk membantu dan memudahkan dalam proses perancangan. Dengan menggunakan teknik metode desain ini dapat membantu dalam merencanakan, mengatur, mengontrol, dan mengevaluasi *database development project*. Proses dalam metodologi perancangan dibagi menjadi tiga tahap yaitu *Conceptual Database Design (CDD)*, *Logical Database Design (LDD)*, dan *Physical Database Design (PDD)*.

1. *Conceptual Database Design (CDD)*

Adalah proses membangun suatu model berdasarkan informasi yang digunakan oleh perusahaan atau organisasi, tanpa pertimbangan perencanaan fisik. adapun beberapa tahapan dari *conceptual database design* yaitu:

- a. Membangun model lokal konseptual data untuk setiap tampilan
- b. Mengidentifikasi jenis entitas
- c. Mengidentifikasi jenis hubungan
- d. Mengidentifikasi dan mengasosiasikan atribut dengan tipe entitas atau hubungan
- e. Menentukan *domain* atribut
- f. Menentukan calon dan atribut *primary key*
- g. Mempertimbangkan penggunaan konsep pemodelan *enhanced* (langkah opsional)

2. *Logical Database Design (LDD)*

Adalah proses pembuatan suatu model informasi yang digunakan pada perusahaan berdasarkan pada model data yang spesifik, tetapi tidak tergantung dari *Database Management System (DBMS)* yang khusus dan pertimbangan fisik yang lain yaitu membangun dan memvalidasi model lokal data logis untuk setiap tampilan dengan beberapa tahapan yaitu:

- a. Menghapus fitur yang tidak kompatibel dengan model relasional (langkah optimal)
- b. Menurunkan hubungan untuk model data lokal logis
- c. Memvalidasi hubungan menggunakan normalisasi
- d. Memvalidasi hubungan terhadap transaksi pengguna
- e. Mendefinisikan batasan integritas

3. *Physical Database Design* (PDD)

Adalah suatu proses untuk menghasilkan gambaran dari implementasi basis data pada tempat penyimpanan, menjelaskan dasar dari relasi, organisasi *file* dan *indeks* yang digunakan untuk efisiensi data dan menghubungkan beberapa *integrity constraints* dan tindakan keamanan dan Menerjemahkan model *global* data logis untuk target DBMS dengan beberapa tahapan yaitu:

- a. Desain dasar hubungan
- b. Desain representasi dari data awal
- c. Desain kendala perusahaan

2.2.3 *Requirement Collection and Analysis*

Menurut (Connolly, 2010), *Requirement Collection and Analysis* adalah proses mengumpulkan data dan menganalisa informasi tentang bagian dari organisasi yang akan didukung oleh system basis data dan memakai informasi ini untuk mengidentifikasi kebutuhan dari sistem baru yang akan dirancang.

Beberapa teknik yang digunakan dalam pengumpulan data dan analisis ini adalah *fact-finding techniques* yang terdiri dari:

1. **Memeriksa Dokumentasi**
Menganalisis data-data yang telah dikumpulkan dari perusahaan untuk mengetahui system yang sedang berjalan
2. **Wawancara**
Proses pengumpulan informasi berupa fakta-fakta yang dilakukan dengan tanya jawab dengan narasumber.

3. Observasi

Metode yang dilakukan dengan cara mengamati langsung jalannya sistem dalam perusahaan untuk mendapatkan informasi yang lebih detail.

4. Penelitian

Mencari informasi yang berhubungan menggunakan jurnal-jurnal, buku referensi dan sumber internet.

5. Kuisisioner


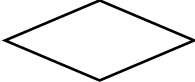

Pengumpulan data dengan menyediakan pertanyaan-pertanyaan yang berhubungan dengan proses bisnis. Ada dua tipe kuisisioner yaitu *open-ended question* dan *close-ended question*.

2.2.4 Entity Relationship Diagram (ERD)

Menurut (Connolly, 2010), *Entity Relationship Diagram* (ERD) digunakan untuk menggambarkan struktur *logical database* dalam bentuk diagram ERD, serta menyediakan cara yang sederhana dan mudah untuk memahami bagian berbagai komponen dalam desain *database*.

ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Notasi yang digunakan dalam komponen ERD dapat dilihat pada Tabel di berikut:

Tabel 2.1 Tabel Notasi ERD

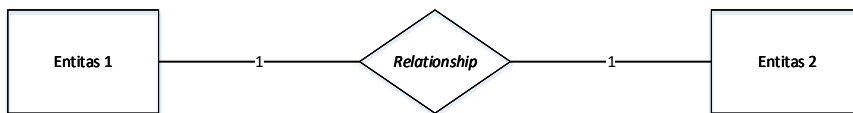
Notasi	Keterangan
	<p>Entity, adalah benda yang memiliki identifikasi yang berbeda. <i>Entity</i> dapat digambarkan sebagai yang berisi nama dari <i>Entity</i> tersebut</p>
	<p>Relationship, adalah hubungan antara <i>Entity</i>. <i>Entity</i> merupakan pengikut dari <i>relationship</i>. <i>Relationship</i> dapat digambarkan dalam bentuk belah ketupat yang mana dari relasi tersebut.</p>
	<p>Artibut, adalah sifat karakteristik deskriptif suatu entitas. Setiap <i>property</i> atau atribut mempunyai <i>key</i> diantaranya <i>primary key</i> (PK) dan <i>foreign key</i> (FK), yang bertujuan untuk menunjukkan hubungan antara kedua entitas.</p>

Notasi	Keterangan
_____	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Dalam ERD (*Entity Relationship Diagram*) terdiri dari beberapa hubungan yaitu *one-to-one*, *one-to-many*, atau *many-to-many*.

- a. Hubungan *one-to-one* (1 atau 1...1)

Hubungan dimana setiap *Entity* yang ada hanya dapat mempunyai maksimal 1 (satu) hubungan dengan *Entity* yang lain.



Gambar 2.1 *One-to-One Relationship*

- b. Hubungan *one-to-many* (1...*)

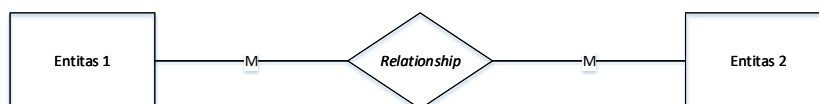
Hubungan dimana setiap *Entity* yang dapat mempunyai satu atau lebih dari satu hubungan dengan *Entity* yang lain.



Gambar 2.2 *One-to-Many Relationship*

- c. Hubungan *many-to-many* (*...*)

Hubungan dimana setiap *Entity* dapat mempunyai lebih dari satu relasi dengan *Entity* lainnya.



Gambar 2.3 *Many-to-Many Relationship*

2.2.5 Normalisasi

Menurut (Connoly, 2010), normalisasi merupakan suatu teknik untuk menghasilkan sekumpulan hubungan dengan property yang diinginkan, yang memberikan kebutuhan data terhadap suatu perusahaan.

Tujuan dari normalisasi adalah sebagai berikut:

- a. Meminimalkan jumlah atribut yang diperlukan untuk mendukung kebutuhan data dari suatu perusahaan.
- b. Untuk memperoleh atribut yang bersifat *functional dependencies*.
- c. Untuk menghilangkan data yang bersifat *redundancy* pada tiap atribut.

Menurut Connoly (2010, p430), Terhadap beberapa bentuk normalisasi pada *database* yaitu:

1. *Unnormalized Normal Form* (UNF)

Unnormalized Normal form (UNF) merupakan sebuah tabel yang mengandung satu atau lebih *repeating group*. Berikut adalah contoh dari bentuk normalisasi UNF ditunjukkan pada tabel 2.2.

Tabel 2.2 Contoh *Normalisasi UNF*

No Nota	Tanggal	Jaminan	No Anggota	Nama	Telepon	Kode Buku	Judul	Jenis	Biaya Sewa	ID Petugas	Nama Petugas
210	8/10/2001	KTP	1001	Doni	82333444000	S1	Soekarno	Sejarah	20000	K01	Muslim
						O1	David Beckham	Olahraga	15000		
						K3	Kungfu Panda	Komik	10000		

2. *First Normal Form* (1NF)

First Normal Form (1NF) merupakan sebuah relasi dimana setiap potongan baris dan kolom mengandung satu dan mungkin hanya satu nilai, dan proses untuk mengubah tabel UNF ke dalam *First Normal Form* (1NF) adalah dengan cara harus diidentifikasi dan menghilangkan bagian yang mengandung *repeating group* pada tabel.

Ada dua pendekatan untuk menghilangkan perulangan kelompok (*repeating group*) dari tabel yang belum dinormalisasikan, yaitu:

- a. Dengan memasukan data kedalam kolom kosong dari baris yang berisi perulangan data. Dengan kata lain, kita mengisi yang kosong dengan

duplikat data yang tidak diulang, yang diinginkan. Pendekatan ini umumnya ditunjuk sebagai *'flattening'* pada tabel.

- b. Dengan menempatkan perulangan data, sepanjang dengan sebuah salinan atribut kunci yang asli ke dalam sebuah relasi yang terpisah. Terkadang tabel yang belum dinormalisasi mungkin berisi lebih dari satu perulangan kelompok. Pada kasus seperti ini, pendekatan dapat diulang sampai tidak ada lagi perulangan yang terjadi. Sebuah set relasi dapat berada pada 1NF jika tidak terdapat perulangan kelompok (*repeating group*).

Berikut adalah contoh bentuk normalisasi 1NF ditunjukkan pada tabel 2.3.

Tabel 2.3 Contoh Normalisasi 1NF

No Nota	Tanggal	Jaminan	No Anggota	Nama	Telepon	Kode Buku	Judul	Jenis	Biaya Sewa	ID Petugas	Nama Petugas
210	8/10/2001	KTP	1001	Doni	82333444000	S1	Soekarno	Sejarah	20000	K01	Muslim
210	8/10/2001	KTP	1001	Doni	82333444000	O1	David Beckham	Olahraga	15000	K01	Muslim
210	8/10/2001	KTP	1001	Doni	82333444000	K3	Kungfu Panda	Komik	10000	K01	Muslim

3. *Second Normal Form (2NF)*

Second Normal Form (2NF) dapat dihasilkan dengan cara melihat apakah ada atribut yang bukan merupakan *primary key* dapat merupakan fungsi dari sebagian *primary key (partial dependence)*.

Dalam bentuk normal kedua setiap atribut yang bergantung secara parsial harus dipisahkan. Bentuk normal akan diperoleh bila setiap atribut yang bukan merupakan *primary key* dari suatu tabel secara penuh yang merupakan *functional dependence* dari *primary key* itu. Berikut adalah contoh bentuk normalisasi 2NF, ditunjukkan pada tabel 2.4.

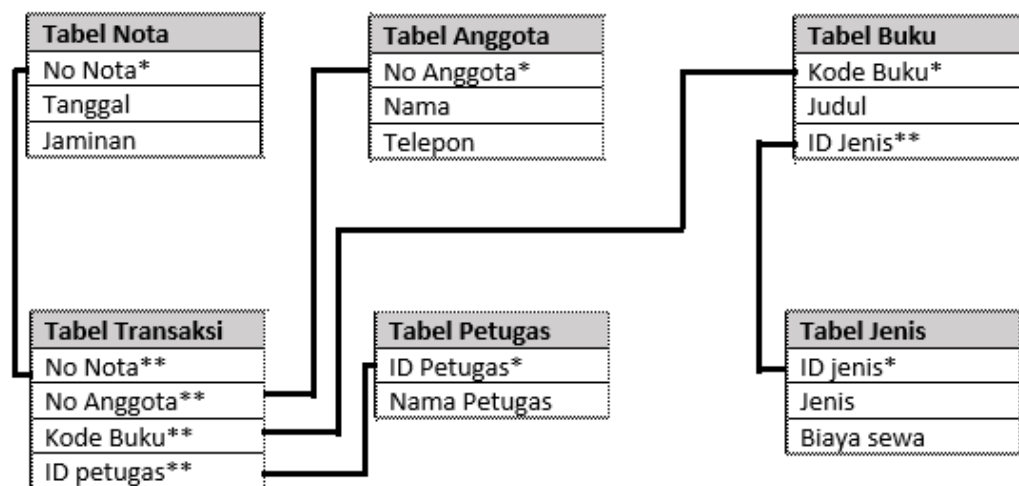
Tabel 2.4 Contoh Normalisasi 2NF

Tabel Nota	Tabel Anggota	Tabel Buku	Tabel Petugas	Tabel Transaksi
No Nota	No Anggota	Kode Buku	ID Petugas	No Nota
Tanggal	Nama	Judul	Nama Petugas	No Anggota
Jaminan	Telepon	Jenis		Kode Buku
		Biaya Sewa		ID petugas

4. *Third Normal Form (3NF)*

Third Normal Form (3NF) akan secara langsung dilakukan pengujian dengan cara melihat apakah terdapat atribut bukan *key* yang bergantung fungsional terhadap atribut yang bukan *key* yang lain atau disebut (*transitive dependence*). Dengan cara yang sama, maka setiap *transitive dependence* harus dipisahkan. *Third Normal Form (3NF)* dapat dikatakan sudah normal apabila anomali yang ada di dalamnya sudah tidak ada, pada kasus tertentu normalisasi dilakukan sampai BCNF. Berikut adalah contoh normalisasi bentuk 3NF ditunjukkan pada gambar 2.5.

Tabel 2.5 Contoh Normalisasi 3NF



2.2.6 MySQL

(Pratama, 2013) MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License (GPL)*, tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka

MySQL didistribusikan sebagai perangkat lunak, di bawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user*

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data

MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed* atau *unsigned* integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

7. Keamanan

MySQL memiliki beberapa lapisan keamanan seperti level subnet mask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, UNIX soket (UNIX), atau Named Pipes (NT).

10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar muka

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*)

12. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABEL*, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.7 PHPMysqlAdmin

Menurut (Sibero, 2011), “*phpMyAdmin* adalah aplikasi *web* yang dibuat oleh *phpMyAdmin*. *Net phpMyAdmin* digunakan untuk administrasi *database MySQL*”. Program ini digunakan untuk mengakses *database MySQL*. Perintah untuk membuat *table* dapat menggunakan *form* yang sudah tersedia pada *phpMyAdmin* atau dapat langsung menulis *script* pada menu *sql*. *phpMyAdmin* dijalankan dengan cara menetik <http://atauataulocalhostatauphpmysqladmin> pada *web browser*.