

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Tinjauan pustaka dilakukan berdasarkan pada penelitian terdahulu, berikut pemaparan beberapa kajian penelitian :

(C Wibowo, A. Angelia, A.Natalia Y. Oenzil (2013)), dalam penelitian yang berjudul Analisis dan Perancangan Sistem Basis Data Untuk Kegiatan Operasional Akademik pada SMP Darma Satria Persada, merupakan penelitian yang bertujuan untuk melakukan analisis terhadap sistem operasional akademik yang diterapkan pada SMP Darma Satria Persada dan melakukan perancangan sistem basis data berupa aplikasi desktop sebagai sarana pengaturan operasional akademik sekolah. Metode penelitian yang digunakan adalah metode pengumpulan data dengan studi pustaka, observasi dan wawancara dan metode perancangan *database* dengan menggunakan metode DBLC (*Database Life Cycle*) yang terdiri dari perancangan konseptual, perancangan logikal, dan perancangan fisik.

(Viriya Adithana L.P. , Anton Wijaya, dan Ali Irawan (2006)), dalam penelitian yang berjudul Analisis dan Perancangan Sistem Basis Data Pendidikan pada Lembaga Musik Cantata, merupakan penelitian yang bertujuan merancang *database* pada Lembaga Musik Cantata yang bergerak di bidang pendidikan music. Metodologi yang digunakan berdasarkan pada *Database Application Lifecycle* (DBLC) Penelitian dimulai dengan mengadakan analisis kebutuhan informasi perusahaan, perancangan basis data konseptual, logikal, dan fisik, memilih *DBMS* yang akan digunakan, dan implementasi. Penelitian telah menghasilkan rancangan *database* yang terdiri dari rancangan konseptual, logikal, dan fisik.

(Tania Liana, Joko Ngalimin (2012)), dalam penelitian yang berjudul Analisis dan Perancangan Sistem Basis Data *E-learning* Berbasis *Web* pada Sekolah Menengah Atas (SMA) Katolik Ricci, merupakan penelitian yang bertujuan untuk melakukan analisis terhadap *system* pendidikan yang diterapkan oleh SMA Katolik Ricci II dan melakukan perancangan *system* basis data berupa

aplikasi *e-learning* yang berbasis web sebagai sarana pembelajaran siswa dan guru yang bersifat tidak terikat oleh waktu dan tempat. Metode penelitian yang digunakan adalah metode pengumpulan data dengan cara metode kuesioner dan studi pustaka, metode perancangan *database* dengan menggunakan metode DBLC (*database life cycle*) yaitu terdiri dari perancangan konseptual, perancangan logikal dan perancangan fisik.

Dari ketiga penelitian di atas tahapan yang digunakan dalam perancangan *database* adalah Pengumpulan dan Analisis Data dengan metode studi pustaka, observasi, kuesioner, dan wawancara, perancangan *database* konseptual, perancangan *database* logikal, dan perancangan *database* fisik. Sedangkan dalam penelitian ini tahapan yang digunakan dalam perancangan *database* adalah pengumpulan dan analisa data (dengan data sekunder yaitu melalui *website* yang merujuk pada *e-learning* dan ujian *online*), perancangan *database* konseptual (perancangan *Entity Relationship Diagram*), perancangan *database* logikal (perancangan *Relational Database*), perancangan *database* fisik (struktur-struktur penyimpanan pada *file-file database*). Selain merancang *database*, penelitian ini juga melakukan pengujian pada *database* apakah *database* mengalami anomali atau tidak, dengan dilakukan pengujian sehingga dapat diperoleh *database* dengan data yang efisien.

## **2.2 Landasan Teori**

### **2.2.1 MS SQL Server**

Menurut (Agus, 2013:11), MS SQL Server adalah salah satu produk *Relational Database Management System* (RDBMS) populer yang berfungsi sebagai relasi *database* dalam sebuah program aplikasi. *Microsoft sql server* mendukung *sql* sebagai bahasa pemrograman *query*, seperti yang kita ketahui *sql* merupakan bahasa standar internasional untuk proses *query database*. Dari *sql* ini sudah banyak sekali digunakan aplikasi pada hampir semua aplikasi baik *e-commerce* pendidikan, organisasi, pemerintahan, dan lain sebagainya.

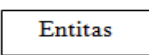

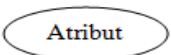

### 2.2.2 Unified Modeling Language (UML)

Menurut (Nugroho, 2009:4), *Unified Modeling Language (UML)* adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metode lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP).

### 2.2.3 Entity Relationship Diagram (ERD)

Menurut (Sutanta, 2011:91), “*Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek.” *Entity Relationship Diagram (ERD)* digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram (ERD)* didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek-obyek dasar tersebut. Penggunaan *Entity Relationship Diagram (ERD)* relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity Relationship Diagram (ERD)* berguna untuk memodelkan sistem basis data yang nantinya akan dikembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data di dalamnya.

Adapun simbol-simbol ERD dapat dilihat pada Gambar 2.1:

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
	Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut.

**Gambar 2. 1** Simbol-simbol ERD

### 2.2.4 Entity Types

Menurut (Connolly dan Begg, 2005:343), tipe entitas sekumpulan objek dengan *property* yang sama yang diidentifikasi oleh perusahaan dan memiliki keberadaan yang bebas atau tidak tergantung. *Entity Occurance* adalah suatu objek dari tipe entitas yang dapat diidentifikasi secara unik. Tipe entitas dibedakan menjadi dua yaitu:

- *Strong Entity type* yaitu tipe entitas yang keberadaannya tidak tergantung kepada tipe entitas lain.
- *Weak Entity Type* yaitu entitas yang keberadaannya bergantung kepada tipe entitas yang lain.

### 2.2.5 Relationship Types

Menurut (Connolly dan Begg, 2005: 346), tipe *relationship* atau hubungan adalah suatu kumpulan atas asosiasi yang memiliki arti di antara tipe entitas. Tiap-tiap tipe *relationship* diberikan nama yang menjelaskan fungsinya masing-masing. Derajat dari tipe *relationship* (hubungan) adalah banyaknya tipe entitas yang berpartisipasi di dalam suatu *relationship*.

### 2.2.6 Attributes

Menurut (Connolly dan Begg, 2005: 350), Merupakan sifat-sifat (*property*) dari sebuah entitas atau *relationship type*. Contohnya: sebuah entitas Staff digambarkan oleh atribut *staffNo*, *Name*, *name*, *position*, dan *salary*. *Attribute Domain* adalah himpunan nilai yang diperbolehkan untuk satu atau lebih atribut. Macam-macam atribut :

- a. *Simple Attribute*, yaitu atribut yang terdiri dari satu komponen tunggal dengan keberadaan yang independen dan tidak dapat dibagi menjadi bagian yang lebih kecil lagi. Dikenal juga dengan nama *Atomic Attribute*.
- b. *Composite Attribute*, yaitu atribut yang terdiri dari beberapa komponen, dimana masing-masing komponen memiliki keberadaan yang independen. Misalkan *Attribute Address* dapat terdiri dari *Street*, *City*, *PostCode*.

- c. *Single-valued Attribute*, yaitu atribut yang mempunyai beberapa nilai untuk setiap kejadian. Misal entitas *Branch* memiliki satu nilai untuk atribut *branchNo* pada setiap kejadian.
- d. *Multi-valued Attribute*, yaitu atribut yang mempunyai beberapa nilai untuk setiap kejadian. Misalnya entitas *Branch* memiliki nilai untuk atribut *telpNo* pada setiap kejadian.
- e. *Derived Attribute*, yaitu atribut yang memiliki nilai yang dihasilkan dari satu atau beberapa atribut lainnya, dan tidak harus berasal dari satu entitas.

### 2.2.7 Key

Menurut (Connolly dan Begg, 2005:352), *Keys* memiliki peran yang sangat penting untuk menghubungkan satu obyek dengan obyek yang lainnya. *Keys* diletakkan pada suatu atribut yang telah ditentukan kedudukannya, agar dapat dihubungkan dengan atribut pada entitas yang lain. Beberapa jenis *keys* yang biasa digunakan antara lain:

#### 1. *Candidate Key*

*Candidate key* yaitu himpunan atribut minimal yang secara unik mengidentifikasi tiap-tiap keberadaan suatu tipe entitas.

#### 2. *Primary Key*

*Primary key* yaitu *candidate key* yang dipilih secara unik untuk mengidentifikasi tiap-tiap keberadaa suatu tipe entitas.

#### 3. *Foreign Key*

*Foreign Key* adalah sebuah kumpulan *field* dalam satu relasi yang digunakan untuk menunjuk ke suatu baris pada relasi yang lain (harus berkorespondensi dengan *primary key* pada relasi yang kedua).

#### 4. *Alternate Key*

*Alternate key* yaitu *candidate key* yang tidak terpilih menjadi *primary key*.

### 2.2.8 Structural Constraints

Menurut (Connolly dan Begg, 2005: 356), *multiplicity* adalah jumlah dari kejadian yang mungkin dari sebuah tipe entitas yang berhubungan kepada sebuah kejadian tunggal dari tipe entitas yang terasosiasi melalui *relationship* (hubungan)

tertentu. Derajat yang bisa digunakan untuk *relationship* (hubungan) adalah *binary*.

*Binary relationship* terdiri atas:

### 1. *One to One (1:1) Relationship*

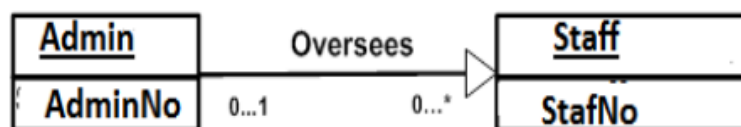
Setiap *relationship* mewakili hubungan antara satu kejadian entitas Staff dan suatu kejadian entitas Branch. Dengan kata lain, seorang staf dapat mengatur nol atau suatu *branch* dan setiap *branch* diatur seorang staf.



Gambar 2. 2 Relasi *One to One*

### 2. *One to Many (1:\*) Relationship*

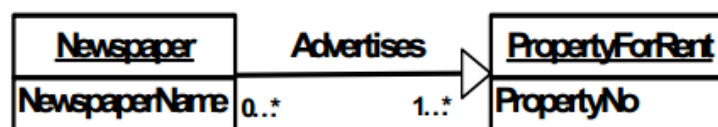
Setiap *relationship* mewakili hubungan antara satu kejadian entitas Admin dan satu kejadian entitas Staff. Dengan kata lain, seorang admin dapat mengawasi nol atau lebih staf dan sebuah *staf* dapat diawasi oleh nol atau satu admin.



Gambar 2. 3 Relasi *One to Many*

### 3. *Many to Many (\*:\*) Relationship*

Setiap *relationship* mewakili hubungan antara satu kejadian entitas Newspaper dan satu kejadian entitas PropertyForRent. Dengan kata lain, Newspaper mengiklankan satu atau lebih *property* dan satu *property* diiklankan pada nol atau lebih *newspaper*.



Gambar 2. 4 Relasi *Many to Many*

### 2.2.9 Metode Perancangan Basis Data

Menurut (Connolly dan Begg, 2005:438-441), perancangan basis data dibagi menjadi tiga tahap utama, yaitu:

1. Perancangan Basis Data Konseptual
2. Perancangan Basis Data Logikal
3. Perancangan Basis Data Fisikal

Berikut penjelasan dari setiap perancangannya:

#### A. Perancangan Basis Data Konseptual

Proses pembentukan model dari informasi yang digunakan dalam perusahaan, bersifat independen dari keseluruhan aspek fisik (Connolly dan Begg, 2005:439). Pertimbangan fisik yang dimaksud meliputi DBMS yang akan digunakan, program aplikasi, Bahasa pemrograman, *platform* perangkat keras, dan pertimbangan lainnya.

Pada konseptual basis data desain ini terdiri dari langkah-langkah sebagai berikut:

- a. Mengidentifikasi tipe entitas.

Tipe entitas dapat dikendalikan dengan mengidentifikasi kata benda atau frase kata benda pada spesifikasi kebutuhan *user*, objek besar seperti orang (*people*), tempat (*place*), benda (*thing*), atau konsep (*concept*). Alternatif lain adalah dengan mencari objek yang keberadaannya bebas.

- b. Mengidentifikasi tipe relasi.

Pada tahapan ini bertujuan untuk mengidentifikasi relasi yang telah diidentifikasi sebelumnya. Tipe relasi diidentifikasi dengan mencari kata kerja atau suatu kata yang berhubungan dengan kata kerja.

- c. Mengidentifikasi dan menghubungkan atribut dengan entitas atau tipe entitas.

Pada tujuannya untuk menghubungkan atribut dengan entitas dan tipe relasi yang tepat. Atribut yang dimiliki oleh setiap entitas dan relasi harus memenuhi karakteristik atribut yaitu *simple* atau *composite*, *single* atau *multi-valued attribute*, dan *derived attribute*.

d. Menentukan domain atribut

Domain adalah sekumpulan nilai dimana satu atau lebih atribut memperoleh nilainya. Contoh menentukan domain pada atribut JenisKelamin di entitas pegawai adalah dengan 'L' atau 'P'.

e. Menentukan *candidate* dan *primary key* dari atribut

*Candidate key* adalah *set* atribut minimal dari sebuah entitas yang secara unik mengidentifikasi setiap kemunculan dari entitas tersebut. *Candidate key* dapat diidentifikasi lebih dari satu, tetapi harus dipilih satu sebagai *primary key*, sedangkan *candidate key* yang lain disebut *alternate key*.

f. Mempertimbangkan penggunaan konsep model yang lebih tinggi (*enhanced modeling concepts*) (*optional*)

Tahap ini bersifat optimal, apakah akan digunakan pengembangan dari entity model dengan menggunakan *enhanced modeling concept*, seperti spesialisasi atau generalisasi.

g. Memeriksa redudansi pada model

Tahapan ini memeriksa model data konseptual lokal apakah terjadi duplikasi atau tidak.

h. Melakukan validasi model data konseptual local terhadap transaksi pengguna.

Memeriksa model yang telah dihasilkan apakah mendukung transaksi pada *view*.

## B. Perancangan Basis Data Logikal

Proses membangun model informasi yang digunakan dalam sebuah *enterprise* yang didasarkan pada data model spesifik, dan terbebas dari DBMS dan semua pertimbangan fisik. Pada desain logical basis data terdiri dari langkah-langkah sebagai berikut:

a. Menghilangkan hal-hal yang tidak sesuai dengan model *relational* (*optional*)

Model data konseptual lokal yang telah ada dapat mengandung struktur yang tidak dapat dimodelkan oleh DBMS konvensional, oleh



karena itu pada tahap ini dilakukan perubahan menjadi bentuk yang lebih mudah ditangani oleh *system* ini.

b. Mendapatkan relasi untuk model data logikal

Membentuk relasi dari model data logical local untuk mempresentasikan relasi antar entitas dengan atribut yang telah diidentifikasi.

c. Melakukan validasi relasi menggunakan normalisasi

Normalisasi digunakan untuk meningkatkan model yang telah terbentuk agar duplikasi data yang tidak diperlukan dapat dihindari.

d. Melakukan validasi relasi dengan transaksi *user*

Bertujuan untuk memastikan bahwa relasi-relasi pada local logikal data model mendukung transaksi-transaksi yang dibutuhkan oleh pengguna, seperti terinci spesifikasi kebutuhan pengguna.

e. Memastikan *integrity constraints*

*Integrity constraints* adalah batasan-batasan yang harus ditentukan untuk melindungi basis data agar tetap konsisten (Connolly dan Begg, 2005:474).

### C. Perancangan Basis Data Fisikal

Proses memproduksi sebuah deskripsi dari implementasi basis data dalam secondary storage, yang menjelaskan relasi dasar, organisasi file, dan membuat indeks untuk mendapatkan akses yang efisien ke data, serta setiap *integrity constraints* yang saling berhubungan dan juga pengukuran sekuriti. Pada desain fisikal basis data ini terdiri dari langkah-langkah sebagai berikut:

a. Menjelaskan model data logical global untuk DBMS yang akan digunakan.

Pada tahap ini akan dihasilkan suatu skema basis data relasional dari model data logikal global yang dapat diimplementasikan ke dalam DBMS yang akan digunakan. Mendesain gambaran fisik dari basis data.

b. Mendesain gambaran fisik dari basis data

Menentukan organisasi *file* yang akan digunakan dan indeks untuk menghasilkan performa yang diinginkan, serta memperkirakan *disk storage* yang diperlukan untuk menggunakan *system* basis data.

### 2.2.10 Normalisasi

Normalisasi adalah suatu teknik untuk menghasilkan himpunan relasi dengan atribut-atribut yang diinginkan berdasarkan kebutuhan-kebutuhan data suatu organisasi (Connolly dan Begg, 2005:388).

Tujuan dilakukannya normalisasi:

1. Untuk mencegah kemungkinan terjadinya redundansi data. Yaitu suatu bentuk kumpulan data dimana data-data yang sudah ada di simpan secara berulang-ulang sehingga mempersulit proses dan susah dipahami. Selain itu juga mengakibatkan pemborosan kapasitas memori dan penyimpanan. Ada 3 macam *update anomalies*, menurut (Connolly dan Begg, 2005:391) yaitu:

*a. Insertion Anomalies*

Penambahan data pada suatu tabel akan mempengaruhi tabel lain yang berelasi dengan tabel tersebut.

*b. Deletion Anomalies*

Anomali yang berkaibat jika kita mendelete sebuah *tupple* pada satu tabel maka akan mempengaruhi perubahan pada tabel yang berelasi dengannya.

*c. Modification Anomalies*

Jika kita ingin mengubah nilai satu *attributes* dari sebuah *tupple* pada satu tabel maka kita harus mengubah informasi tersebut yang ada di tabel lain agar *database* tetap konsisten.

2. Untuk memvalidasi apakah kebutuhan bisnis sebenarnya sudah sesuai dengan bentuk relasinya. Tahapan normalisasi menurut (Connolly dan Begg, 2005:401), adalah sebagai berikut:

*a. Unnormalization Form (UNF)*

Sebelum membahas bentuk normal yang pertama, kita mendefinisikan normal form awal yaitu *Unnormalized form (UNF)*. UNF adalah sebuah tabel yang berisi satu atau lebih dari grup-grup yang berulang. (Connolly dan Begg, 2005:403).

b. First Normal Form (1NF)

*First Normal Form* (1NF) adalah sebuah relasi yang mana pemisahan dari setiap baris dan kolom berisi 1 dan hanya 1 nilai. (Connolly dan Begg, 2005:403) *Second Normal Form* (2NF)

*Second Normal Form* (2NF) adalah sebuah relasi yang terdapat pada *First Normal Form* dan setiap atribut selain *primary key* sepenuhnya bergantung fungsinya pada *primary key*. (Connolly dan Begg, 2005:407) Normalisasi dari hubungan 1NF ke 2NF melibatkan pemindahan sebagai ketergantungan. Jika sebuah ketergantungan sebagian terjadi, maka kita harus menyingkirkan atribut ketergantungan sebagian dari relasi dengan menempatkan mereka pada sepanjang relasi yang baru dengan Salinan dari factor penentu mereka.

d. *Third Normal Form* (3NF)

*Third Normal Form* (3NF) adalah sebuah relasi yang terdapat pada *First Normal Form* dan *Second Normal Form* dan dimana tidak terdapat atribut yang bukan *primary key* yang bergantung sepenuhnya pada *primary key*. (Connolly dan Begg, 2005:409) Normalisasi dari hubungan 2NF ke 3NF melibatkan ketergantungan secara lengkap. Jika terdapat atribut ketergantungan secara lengkap dari relasi dengan menempatkan atribut pada sepanjang relasi yang baru dari salinan dari faktor penentu mereka.

### 2.2.11 Asesmen dalam Pembelajaran

Asesmen merupakan segala tindakan yang dilakukan oleh guru maupun siswa untuk menilai pencapaian mereka dan menggunakan hasilnya untuk mengubah aktivitas belajar mengajar mereka (Black, et al., 2001).

Untuk menjalankan asesmen yang mampu meningkatkan pembelajaran dengan baik, ada beberapa hal yang harus dipenuhi (Black, et al., 2001):

1. Asesmen harus menghasilkan informasi yang akurat.
2. Feedback yang diberikan kepada siswa lebih baik deskriptif dibanding evaluatif.
3. Siswa diikutsertakan dalam proses asesmen.

Dalam mengikutsertakan siswa dalam proses asesmen, (Stiggins et al., 2007) menjelaskan bahwa siswa perlu untuk mengetahui apa yang seharusnya

mereka tahu, apa yang mereka tahu sekarang, dan bagaimana cara memperkecil jarak antara keduanya. (Stiggins et al., 2007) kemudian menjelaskan bagaimana sebaiknya asesmen dilakukan untuk membantu siswa dalam mengetahui jawaban atas 3 hal tersebut:

1. Memberikan visi dari target pembelajaran dengan jelas dan mudah dipahami.
2. Memberikan contoh dan model mengenai pekerjaan atau hasil yang baik dan buruk.
3. Memberikan *feedback* yang deskriptif secara regular.
4. Mengajarkan siswa untuk menetapkan tujuan dan melakukan asesmen pribadi.
5. Mendesain pelajaran agar fokus pada satu aspek dalam satu waktu.
6. Mengajarkan siswa bagaimana membuat revisi pembelajaran yang terfokus.
7. Mengajak siswa untuk melakukan tracking, refleksi, dan komunikasi mengenai perkembangan pembelajaran mereka secara pribadi.