

BAB 2

LANDASAN TEORI

2.1 Teori Umum

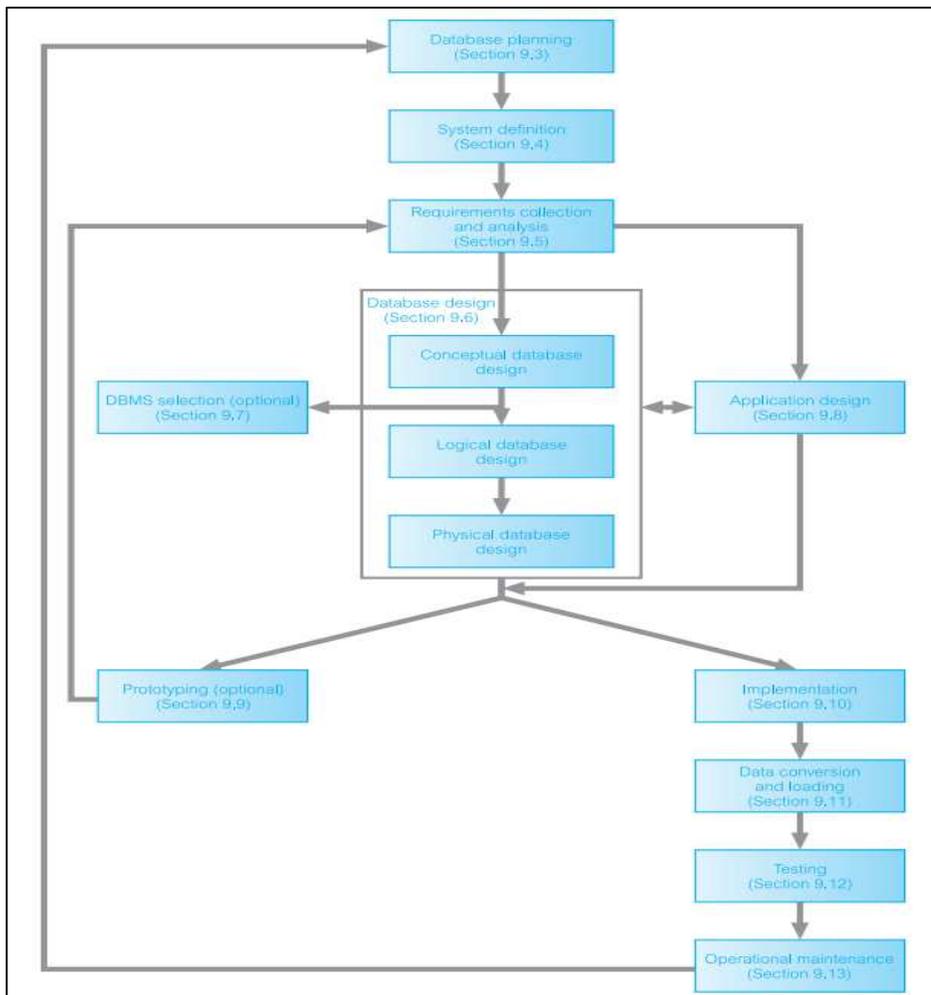
2.1.1 *Database*

Database adalah sekumpulan Data yang terhubung secara logika dan deskripsinya (*metaData*, yaitu Data tentang Data) yang dipakai bersama, dirancang untuk memenuhi kebutuhan informasi suatu organisasi (Connolly, T., and Begg, C. 2015:63). *Database* adalah penyimpanan Data secara besar yang dapat digunakan berkali-kali oleh pengguna. *Database* mengintegrasikan semua Data dan mengurangi duplikasi Data bila dibandingkan dengan penyimpanan Data secara *file-based*. *Database* merepresentasikan entitas-entitas, atribut-atribut, dan hubungan logikal antarentitas. Salah satu keuntungan dari *Database* adalah abstraksi Data, yaitu kita dapat mengubah definisi internal dari sebuah objek tanpa mempengaruhi pengguna dari objek tersebut. Sebelum adanya *Database*, pada umumnya perusahaan menggunakan sistem *file-based*.

Sistem *file-based* adalah sekumpulan program aplikasi yang menjalankan beberapa layanan untuk pengguna akhir, seperti membuat laporan. Setiap program menentukan dan mengatur Data masing-masing (Connolly, T., and Begg, C. 2015:55). Sistem *file-based* memiliki beberapa batasan yaitu, Data yang terpisah dan terisolasi, duplikasi Data, kemandirian Data, format *file* yang tidak cocok, dan *query* yang statis dari program aplikasi. Berdasarkan O'Brien, A. J., and Marakas, M.G. (2010:181), *Database* adalah sekumpulan elemen Data yang saling berhubungan secara logikal dan saling terintegrasi.

2.1.2 *Database System Lifecycle*

Connolly, T., and Begg, C. (2015:347), berpendapat untuk merancang aplikasi sistem *Database* diperlukan tahapan-tahapan yang dinamakan dengan siklus hidup aplikasi *Database system development lifecycle*. Tahapan-tahapan *Database system development lifecycle* adalah sebagai berikut:



Gambar 2.1 Database System Lifecycle

1. Database Planning
2. System Definition
3. Requirements Collection and Analysis
4. Database Design
5. DBMS Selection
6. Application Design
7. Prototyping
8. Implementation
9. Data Conversion and Loading
10. Testing
11. Operational Maintenance

2.1.2.1 Database Planning

Langkah awal yang dilakukan dalam perencanaan pembuatan *Database* adalah mendefinisikan dengan jelas *mission statement* dari sistem *Database*. *Mission statement* mendefinisikan *mission objective* dari sistem *Database*. Mereka yang mendorong proyek *Database* dalam organisasi biasanya menemukan *mission statement*. Sebuah *mission statement* membantu memperjelas tujuan dari sebuah sistem *Database* dan memberikan penjelasan yang lebih jelas agar dapat terciptanya sistem *Database* yang efisien dan efektif. Setelah *mission statement* didefinisikan, aktivitas berikutnya adalah mengidentifikasi *mission objective*. Setiap *mission objective* harus mengidentifikasi tugas tertentu yang mendukung sistem *Database*.

Logikanya ialah jika sistem *Database* mendukung *mission objective*, maka *mission statement* harus dipenuhi. *Mission statement* dan *objective* bisa disertai dengan beberapa informasi tambahan yang menentukan secara umum pekerjaan yang harus dilakukan, sumber daya yang dapat digunakan untuk melakukannya, dan uang untuk membayar untuk itu semua.

2.1.2.2 System Definition

Penentuan *system definition* digunakan untuk mendeskripsikan jangkauan dan batasan dari aplikasi *Database* serta pandangan-pandangan utama para pengguna aplikasi. Sebelum mendesain suatu sistem *Database*, terlebih dahulu melakukan identifikasi batasan-batasan dari sistem yang sedang diselidiki dan bagaimana kaitannya dengan bagian lain dari sistem informasi perusahaan. Termasuk dalam ruang lingkup dan batas sistem *Database* adalah pandangan utama pengguna yang harus didukung oleh *Database*.

2.1.2.3 Requirement Collection dan Analysis

Requirement collection dan *analysis*, merupakan proses mengumpulkan dan menganalisis informasi dari organisasi yang akan didukung oleh aplikasi *Database*, dan informasi ini digunakan untuk

mengidentifikasi kebutuhan pengguna aplikasi terhadap sistem baru (Connolly, T., and Begg, C. 2015:350).

Ada 2 macam pendekatan untuk membuat sebuah keputusan aplikasi *Database* dengan berbagai cara pandang pengguna, yaitu:

1. Pendekatan terpusat (*centralized*) menurut Connolly, T., and Begg, C. (2015:352), kebutuhan untuk setiap sudut pandang pengguna disatukan menjadi satu set kebutuhan untuk aplikasi *Database*. Umumnya pendekatan ini digunakan untuk *Database* yang tidak terlalu rumit.
2. Pendekatan integrasi sudut pandang (*view integration*) menurut Connolly, T., and Begg, C. (2015:350), kebutuhan untuk setiap pandangan pengguna digunakan untuk membangun sebuah model Data-Data terpisah yang merepresentasikan tiap pandangan pengguna sistem. Hasil dari model Data-Data tersebut kemudian disatukan di bagian perancangan *Database*.

2.1.2.4 Database Design

Menurut Connolly, T., and Begg, C. (2015:354), perancangan *Database* merupakan proses pembuatan suatu desain atau rancangan untuk sebuah *Database* yang akan mendukung operasional dan target suatu perusahaan. Berikut tahapan dalam *Database design*, adalah :

2.1.2.4.1 User View

User view mendefinisikan apa yang dibutuhkan dari sistem *Database* berdasarkan perspektif dari peran pekerjaan tertentu (seperti *manager* atau *supervisor*) atau perusahaan daerah aplikasi (seperti pemasaran, personalia, atau kontrol stok). Sebuah sistem *Database* mungkin memiliki satu atau lebih pandangan pengguna. Mengidentifikasi pandangan pengguna suatu aspek penting dari pengembangan sistem *Database* karena membantu untuk memastikan bahwa tidak ada pengguna utama *Database* yang dilupakan ketika mengembangkan persyaratan untuk sistem *Database* baru.

Pandangan pengguna juga sangat membantu dalam pengembangan dari sistem *Database* yang relatif kompleks.

2.1.2.4.2 Conceptual Database Design

Proses konstruksi model dari Data yang digunakan dalam suatu organisasi, yang *independent* dari semua pertimbangan fisik. Konseptual Data model terdiri dari tipe entitas, tipe relasi, *attributes* dan *attribute domains*, *primary keys* dan *alternate keys*, *integrity constraints*. Model Data konseptual didukung oleh dokumentasi, termasuk ER *diagrams* dan kamus Data, yang diproduksi di seluruh pengembangan model. Berikut merupakan tahap dari perancangan *Database* konseptual:

- a. Mengidentifikasi tipe entitas.
- b. Mengidentifikasi tipe hubungan (*relationship*).
- c. Mengidentifikasi dan menghubungkan atribut-atribut dengan tipe entitas atau relasi (*relationship*).
- d. Menentukan *domain* atribut.
- e. Menentukan atribut *candidate*, *primary*, dan *alternate key*.
- f. Mempertimbangkan penggunaan *enhanced modeling concepts*.
- g. Memeriksa model akan adanya redundansi.
- h. Memvalidasi model konseptual lokal terhadap transaksi pengguna.
- i. Meninjau kembali model Data konseptual dengan pengguna.

2.1.2.4.3 Logical Database Design

Proses konstruksi model dari Data yang digunakan dalam suatu organisasi berdasarkan dari sebuah Data model yang spesifik, tapi *independent* dari DBMS tertentu dan juga pertimbangan fisik lainnya. Berikut merupakan tahapan dalam merancang *Database* logikal:

1. Membuat relasi untuk model Data logikal.
2. Melakukan validasi relasi menggunakan normalisasi.
3. Melakukan validasi relasi terhadap transaksi pengguna.
4. Memeriksa batasan-batasan integritas.
5. Meninjau kembali model Data logikal dengan pengguna.
6. Menggabungkan model Data logikal menjadi model Data global (*optional*).
7. Mengecek perkembangan di masa depan.

2.1.2.4.4 Physical Database Design

Proses memproduksi sebuah deskripsi dari implementasi sebuah *Database* pada *secondary storage*. Fase ini mendeskripsikan relasi, organisasi *file*, dan juga *index-index* yang digunakan untuk mencapai akses Data yang efisien, dan semua *integrity constraint* dan *security measures* yang terasosiasi. Tujuan utama dari perancangan *Database* fisik adalah:

- a. Membuat seperangkat *tabel* relasi dan juga *constraint-constraint tabel* relasi tersebut dari informasi yang terdapat pada logikal Data model.
- b. Identifikasi struktur penyimpanan yang spesifik dan metode akses Data untuk mencapai performa yang optimal.
- c. Melakukan desain proteksi keamanan dari sistem.

Berikut merupakan tahapan dalam perancangan *Database* fisik:

1. Menerjemahkan model Data logikal untuk sasaran DBMS, dengan:
 - a. Merancang relasi dasar.
 - b. Merancang representasi dari *derived* Data.
 - c. Merancang *general constraint*.
2. Merancang organisasi *file* dan indeks, dengan:
 - a. Menganalisis transaksi.
 - b. Memilih organisasi *file*.

- c. Memilih indeks.
 - d. Memperkirakan kebutuhan ukuran *disk*.
3. Merancang *user view*
 4. Merancang mekanisme keamanan

2.1.2.5 DBMS Selection

Menurut Connolly, T., and Begg, C. (2015:64), *Database management system* merupakan sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, merancang atau membuat, merawat, dan mengontrol akses ke dalam *Database*.

Pemilihan DBMS yang tepat sangat mendukung aplikasi *Database*. Langkah utama dalam pemilihan DBMS adalah sebagai berikut:

1. Mendefinisikan waktu untuk melakukan studi referensi.
2. Mencatat 2 atau 3 jenis produk yang akan dievaluasi untuk digunakan.
3. Mengevaluasikan produk tersebut.
4. Merekomendasikan produk yang dipilih dan membuat laporan yang mendukung DBMS tersebut.

2.1.2.6 Application Design

Menurut Connolly, T., and Begg, C. (2015:363), *application design* adalah proses merancang *user interface* atau antarmuka pengguna dan program aplikasi yang menggunakan dan memproses *Database*.

Terdapat 2 aspek penting dalam perancangan aplikasi, yaitu:

1. *Transaction Design*

Transaksi merupakan sebuah tindakan atau seperangkat tindakan, yang dilakukan oleh pengguna atau program aplikasi, yang mengakses atau mengubah isi dari *Database*.

Ada tiga jenis transaksi:

a. *Retrieval Transaction*

Digunakan untuk mendapatkan kembali Data untuk ditampilkan dalam suatu laporan.

b. *Update Transaction*

Digunakan untuk memasukkan Data, menghapus Data lama, atau mengubah Data yang sudah ada dalam *Database*.

c. *Mixed Transaction*

Merupakan kombinasi antara *retrieval transaction* dan *update transaction*.

2. *User Interface Design Guidelines*

Sebelum mengimplementasikan sebuah *form* atau laporan, penting bagi kita untuk terlebih dahulu mendesain tampilannya.

Terdapat beberapa pedoman dalam merancang *form* atau laporan, yaitu:

- a. Judul yang bermakna.
- b. Instruksi yang mudah dipahami.
- c. Pengelompokkan yang logis dan kolom yang berurutan.
- d. Visualisasi tata letak yang menarik dari laporan.
- e. Nama *field* yang mudah dipahami
- f. Terminologi dan singkatan yang konsisten.
- g. Warna yang digunakan konsisten.
- h. Ruang dan batas-batas untuk *field* memasukkan Data jelas.
- i. Pergerakan kursor yang mudah.
- j. Perbaikan kesalahan pada masing-masing karakter dan seluruh *field*.
- k. Menampilkan pesan *error* untuk nilai yang tidak dapat diterima.
- l. *Field* pilihan ditandai dengan jelas.
- m. Pesan yang jelas untuk *field*.
- n. Adanya pemberitahuan selesai, ketika pengguna selesai mengisi *field* dari suatu formulir.

2.1.3 *Database Language*

Menurut Connolly, T., and Begg, C. (2015:89), dalam pembuatan *Database* bahasa yang digunakan untuk membuat sebuah *Database* terbagi menjadi 2 bagian, yaitu:

2.1.3.1 Data *Definition Language*

Menurut Connolly, T., and Begg, C. (2015:90), *Data definition language* adalah sebuah bahasa yang memungkinkan *Database administrator* dan pengguna untuk menjelaskan dan menamakan entitas, atribut, dan juga relasi yang dibutuhkan untuk aplikasi, bersama dengan integritas yang berhubungan dan batasan keamanan.

2.1.3.2 Data *Manipulation Language*

Menurut Connolly, T., and Begg, C. (2015:90), *Data manipulation language* adalah sebuah bahasa yang menyediakan seperangkat operasi untuk mendukung operasi dasar manipulasi Data pada Data yang disimpan dalam *Database*.

Operasi manipulasi Data biasanya mencakup:

1. Penambahan Data baru.
2. Modifikasi Data yang tersimpan dalam *Database*.
3. Pengambilan Data yang terdapat pada *Database*.
4. Penghapusan Data dari *Database*.

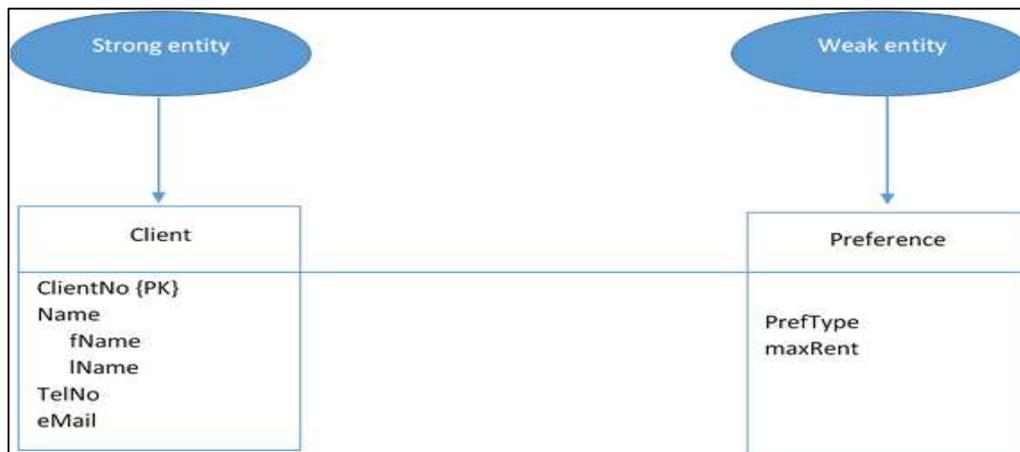
2.1.4 *Entity Relationship Diagram*

Entity relationship diagram merupakan sebuah teknik untuk menggambarkan struktur logikal *Database* dalam bentuk diagram. Pemodelan *entity relationship* merupakan pemodelan Data tingkat tinggi yang termasuk dalam pendekatan *top-down* dalam perancangan *Database*. Pemodelan ini didasarkan pada persepsi dunia nyata yang terdiri atas sekumpulan obek-objek dasar yang disebut dengan entiti, serta *relationship* diantara objek objek tersebut.

2.1.5 *Entity Types*

Entity types didefinisikan sebagai benda atau objek pada dunia nyata yang memiliki properti yang sama. Kesamaan tipe ini dapat dilihat dari atribut yang dimiliki oleh setiap *entity*. Representasi diagramatik dari *entity* adalah digambarkan berupa persegi panjang berlabel nama dari masing-masing *entity*. Tipe entitas terbagi menjadi dua, *strong entity* dan *weak entity*. *Strong entity* adalah tipe entitas yang tidak bergantung pada tipe entitas lain,

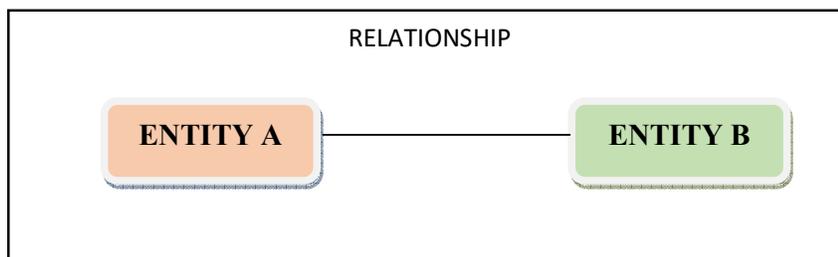
sedangkan *Weak entity* adalah tipe entitas yang bergantung pada tipe entitas lain.



Gambar 2.2 Entity Types

2.1.6 Relationship Types

Relationship types didefinisikan sebagai hubungan yang terjadi antar *entity*. Representasi diagramatik dari *entity* adalah berupa garis lurus yang menghubungkan 2 *entity*, atau lebih yang dikenal dengan istilah *Binary Relationship*.

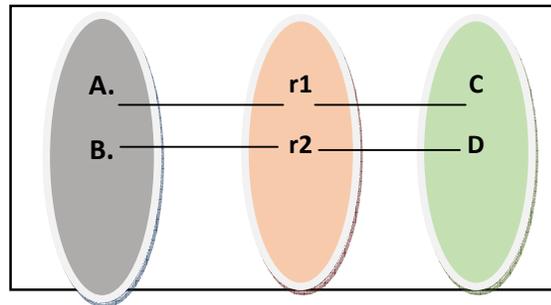


Gambar 2.3 Representasi Diagramatik dan Relationship Types

Terdapat tiga jenis relasi biner, yaitu:

1. Relasi Biner 1:1 (*one to one*)

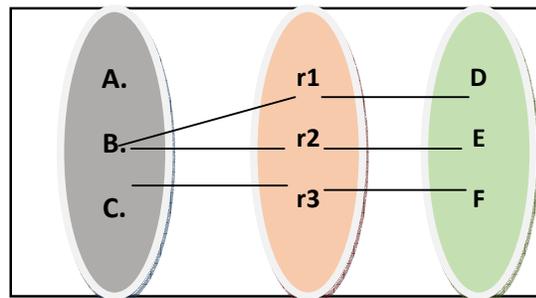
Relasi ini terjadi jika suatu *entity* tunggal berelasi dengan *entity* lainnya.



Gambar 2.4 Relasi Biner 1:1 (*one to one*)

2. Relasi Biner 1:* (*one to many*)

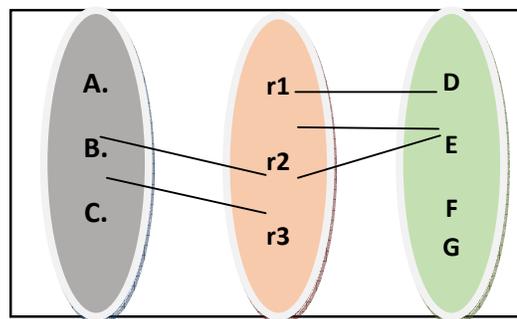
Relasi ini terjadi jika suatu *entity* tunggal berelasi lebih dari satu *entity* lainnya.



Gambar 2.5 Relasi Biner 1:* (*one to many*)

3. Relasi Biner **:*(*many to many*)

Relasi ini terjadi jika suatu *entity* tunggal berelasi dengan banyak *entity* lainnya.



Gambar 2.6 Relasi Biner **:*(*many to many*)

2.1.7 Attribute

Attribute didefinisikan sebagai objek-objek yang ada disetiap *entity*. Suatu *attribute* bisa bernilai tunggal (*single-valued*) atau bernilai jamak (*multi-valued*).

a. *Single-Valued Attribute*

Sebuah atribut yang memiliki nilai tunggal untuk setiap kejadian (baris) dari tipe entitas, contohnya adalah cabang (Connolly, T., and Begg, C. 2015:414).

b. *Multi-Valued Attribute*

Sebuah atribut yang memiliki nilai lebih dari satu untuk setiap kejadian (baris) dari tipe entitas, contohnya adalah nomor telepon (Connolly, T., and Begg, C. 2015:414).

c. *Derived Attribute*

Sebuah atribut yang menggambarkan nilai dari hasil penurunan atribut yang berhubungan, contohnya adalah umur (Connolly, T., and Begg, C. 2015:414).

2.1.8 Key

Key adalah satu gabungan dari beberapa atribut yang dapat membedakan semua basis Data (*row*) dalam tabel. *Key* dalam *Database* berfungsi untuk mengidentifikasi dan menghubungkan satu tabel dengan tabel lain.

a. *Primary Key*

Primary key didefinisikan berdasarkan keunikan untuk mengidentifikasi pada setiap *entity* (Connolly, T., and Begg, C. 2015:415).

b. *Candidate Key*

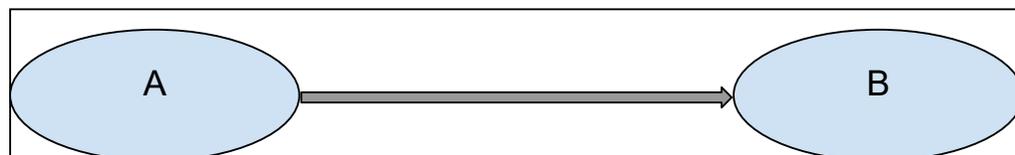
Candidate key didefinisikan sebagai jumlah minimal dari suatu atribut yang mempunyai nilai secara unik untuk mengidentifikasi pada setiap *entity* (Connolly, T., and Begg, C. 2015:415).

c. *Composite Key*

Composite key didefinisikan sebagai *candidate key* yang terdiri dari dua atau lebih pada *attribute* (Connolly, T., and Begg, C. 2015:416).

2.1.9 Normalization

Connolly, T., and Begg, C. (2015:451), mengatakan normalisasi adalah teknik desain *Database* yang dimulai dengan memeriksa hubungan disebut dependensi fungsional antara atribut. Tujuan dari normalisasi adalah untuk mengidentifikasi satu set yang cocok dari hubungan yang mendukung persyaratan Data perusahaan.



Gambar 2.7 Normalization

a. 1NF

Bentuk normal pertama atau *first normal form* (1NF) adalah sebuah relasi dimana persimpangan setiap baris dan kolom memiliki satu dan hanya satu nilai (Connolly, T., and Begg, C. 2015:466).

b. 2NF

Bentuk normal kedua atau *second normal form* (2NF) adalah sebuah relasi yang pada bentuk normal pertama dan setiap atribut yang bukan *primary key* bergantung fungsional secara sebagian sekumpulan *candidate key* (Connolly, T., and Begg, C. 2015:470).

c. 3NF

Bentuk normal ketiga atau *third normal form* (3NF) adalah sebuah relasi yang ada pada bentuk normal pertama dan kedua dan dimana tidak ada

atribut yang bukan *primary key* bergantung secara transitif pada *primary key* (Connolly, T., and Begg, C. 2015:471).

2.1.9.1 Ketergantungan Fungsional

Ketergantungan fungsional menjelaskan hubungan antaratribut dalam sebuah relasi. Ada beberapa jenis ketergantungan fungsional adalah sebagai berikut (Connolly, Begg, 2015:457).

a. Ketergantungan Fungsional Penuh

Ketergantungan fungsional penuh mengindikasikan bahwa jika A dan B adalah atribut dari sebuah relasi, B secara penuh bergantung secara fungsional pada A jika B bergantung secara fungsional bergantung pada A, tetapi bukan pada bagian A yang tepat.

b. Ketergantungan Fungsional Sebagian

Ketergantungan Fungsional Sebagian adalah dimana beberapa atribut pada A dapat dihilangkan tetapi masih ada ketergantungan.

c. Ketergantungan Transitif

Ketergantungan transitif adalah sebuah kondisi di mana A, B, dan C adalah atribut dari relasi dimana jika B bergantung pada A, dan C bergantung pada B, kemudian C secara transitif bergantung pada A melalui B.

2.1.10 Activity Diagram

Activity diagrams model adalah aliran kontrol dari satu aktivitas ke aktivitas lainnya. *Activity diagrams* biasanya merupakan permohonan operasi, langkah dalam suatu bisnis proses, atau keseluruhan bisnis proses. Diagram menunjukkan aliran kontrol dan cabang (*decision*) dapat digunakan untuk menentukan jalur alternatif transisi.

2.1.11 Teori 5W+1H

Teori 5W1H yang disampaikan oleh Pradeep Mahalik adalah metode pengajuan pertanyaan tentang suatu proses atau masalah yang diambil untuk memecahkan permasalahan. Empat dari W (*who, what, where, when*) dan yang H (*how*) digunakan untuk memahami rincian, menganalisis kesimpulan dan penilaian untuk mendapatkan fakta-fakta fundamental dan memandu

pernyataan untuk sampai ke abstraksi. W terakhir (*why*) sering ditanyakan lima kali sehingga seseorang dapat menelusuri untuk sampai ke inti masalah.

5W1H dari six sigma, menjelaskan pendekatan yang harus diikuti dengan memahami dan menganalisis proses, proyek atau masalah untuk perbaikan. Berikut ini adalah garis besar pendekatan 5W1H untuk six sigma (Pradeep Mahalik).

2.2 Teori Khusus

2.2.1 SQL Server Management Studio

SQL *server management studio* adalah sebuah lingkungan terpadu untuk mengakses, mengkonfigurasi, mengelola, dan mengelola semua komponen SQL *server*. *Management studio* menggabungkan sekelompok alat grafis yang luas dengan lima *editor* naskah yang kaya untuk menyediakan akses ke SQL *server* untuk pengembang dan *administrator* dari semua tingkat keahlian. *Management studio* dirancang untuk mengembangkan dan mengelola objek *Database*, dan untuk mengelola dan mengkonfigurasi layanan analisis objek yang sudah ada.

2.2.2 Visual Studio

Visual Studio adalah sebuah kumpulan lengkap dari peralatan pengembangan aplikasi untuk menciptakan aplikasi *web* ASP.NET, servis *web* XML, aplikasi *desktop*, dan aplikasi *mobile*. *Visual basic*, *visual C#* dan *visual C++* menggunakan *integrated development environment* yang sama, yang memungkinkan untuk membagikan peralatan dan mempermudah dalam menciptakan solusi bahasa gabungan. Bahasa-bahasa tersebut menggunakan fungsi dari .NET *framework*, yang menyediakan akses ke teknologi-teknologi utama yang menyederhanakan pengembangan dari aplikasi *web* ASP dan servis *web* XML.

2.2.3 Pengertian Pembelian

Pembelian adalah suatu proses yang dilakukan dalam melaksanakan transaksi antar dua pihak atau lebih (Kamus Besar Bahasa Indonesia).

2.2.4 Pengertian Pengadaan

Pengadaan merupakan proses kegiatan untuk pemenuhan atau penyediaan kebutuhan dan pasokan barang atau jasa dibawah kontrak atau pembelian langsung untuk memenuhi kebutuhan bisnis. Pengadaan dapat mempengaruhi keseluruhan proses arus barang karena merupakan bagian penting dalam proses tersebut (Kamus Besar Bahasa Indonesia).

2.2.5 Daftar Istilah yang Digunakan dalam Penelitian

a. *Coal Bed Methane*

Coal bed methane adalah suatu bentuk gas alam yang berasal dari batu bara (*coal*).

b. *Wellhead*

Wellhead adalah alat yang digunakan dalam pengeboran minyak. *Wellhead* berfungsi untuk menjadi dudukan *christmas tree* dan untuk menggantung casing atau tubing pada suatu sumur. Selain itu fungsi lain *wellhead* adalah untuk mengontrol operasi atau aktivitas dipermukaan sumur yaitu menyekat untuk mencegah kebocoran fluida formasi atau tersembur dipermukaan.

c. *Rig* Pengeboran

Rig pengeboran adalah suatu instalasi peralatan untuk melakukan pengeboran ke dalam *reservoir* bawah tanah untuk memperoleh air, minyak, atau gas bumi, atau deposit mineral bawah tanah. *Rig* pengeboran bisa berada di atas tanah (*on shore*) atau di atas laut atau lepas pantai (*off shore*) tergantung kebutuhan pemakaiannya.

d. *Rock Bit*

Rock bit adalah alat penghancur bebatuan, letaknya dipasang di ujung rangkaian pipa pemboran, adanya putaran dan beban yang diperoleh dari rangkaian pipa bor di atasnya akan menyebabkan mata bor menghancurkan batuan yang terletak dibawah sehingga akan menembus bebatuan tersebut dengan dalam.

e. *Drifting*

Drifting merupakan tehnik yang dilakukan ketika pengecekan pipa atau *casing* dilakukan dengan cara mendorong besi kedalam pipa tersebut sehingga mengetahui apakah pipa tersebut layak pakai atau tidak.

f. *Drill Site Preparation*

Drill site preparation adalah proses persiapan yang dilakukan pada area dimana pemboran akan dijalankan, mulai dari tempat istirahat, persiapan pembangunan jalan untuk transportasi alat berat, dan pos pengamanan.

g. *Tubular Goods*

Tubular goods merupakan benda menyerupai pipa yang akan digunakan untuk aktivitas pemboran.

h. *Production Sharing Contract*

Production sharing contract adalah mekanisme kerjasama pengelolaan migas antara pemerintah dan kontraktor.

i. *Authorization for Expenditure*

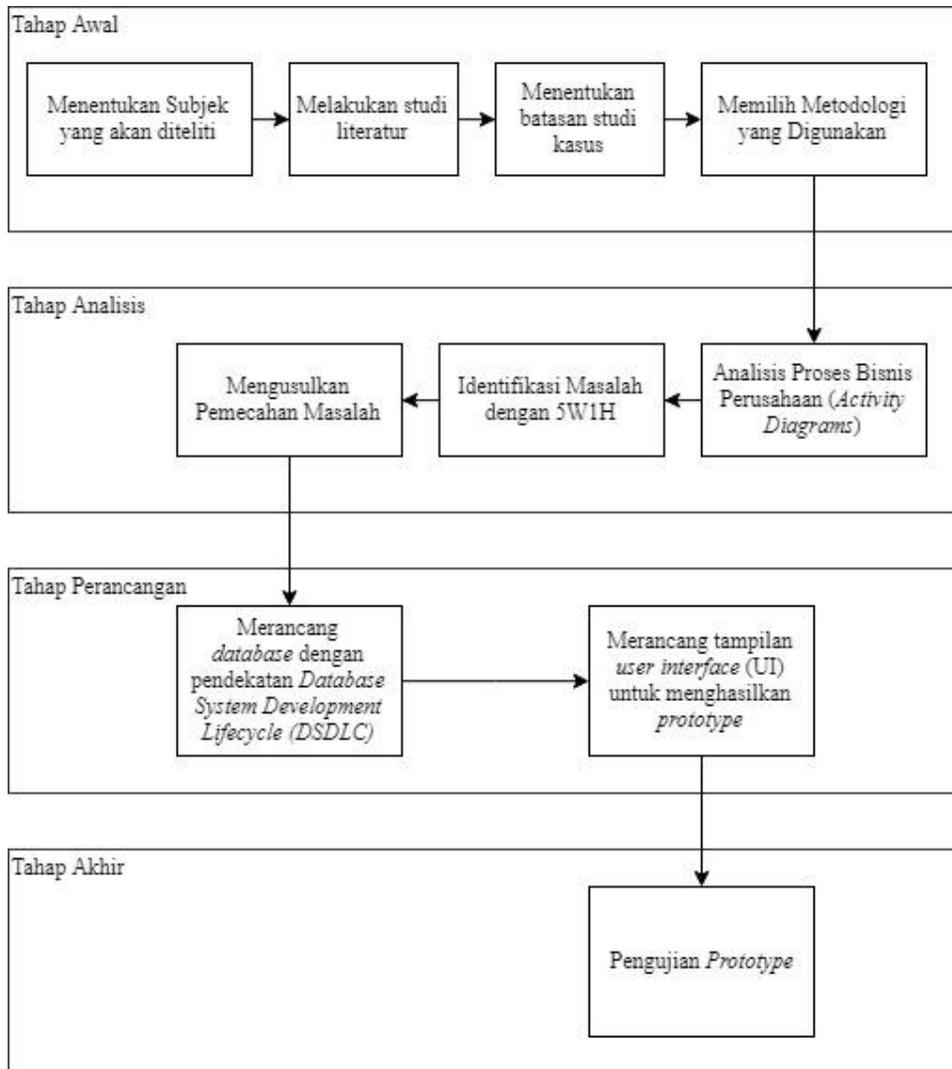
Authorization for Expenditure adalah mekanisme pengendalian pembelanjaan berbasis proyek yang menyediakan informasi bagi BPMIGAS.

j. *Work Program dan Budget*

Work program dan *budget* merupakan usulan rincian rencana kegiatan dan anggaran tahunan dengan mempertimbangkan tentang kondisi, komitmen, efektivitas dan efisiensi pengoperasian kontraktor di suatu wilayah kontrak kerja.

2.3 Kerangka Pikir

Gambar 2.8 menggambarkan kerangka berpikir dalam membuat penulisan ini. Mulai dari tahap awal yaitu menentukan subjek yang akan diteliti, yaitu Dart Energy (Tanjung Enim) Pte Ltd. Kemudian, studi literatur dilakukan untuk mengetahui studi kasus yang sudah dilakukan sebelumnya mengenai perancangan *Database*. Setelah melakukan studi literature, ruang lingkup studi kasus ditentukan yaitu pada proses pencatatan penerimaan dan pengeluaran barang pada gudang Dart Energy dan menentukan metodologi yang akan digunakan untuk merancang *Database*. Pada tahap berikutnya yaitu tahap analisis, analisis proses bisnis perusahaan yang dilakukan. Setelah menganalisa proses bisnis perusahaan maka dilakukan identifikasi masalah dan mengusulkan solusi pemecahan masalah. Tahap berikutnya adalah tahap perancangan *Database* yang meliputi perancangan konseptual, logical dan fisik. Tahap terakhir adalah pengujian *prototype* yang sudah dibangun.



Gambar 2.8 Kerangka Pikir Perancangan Database pada Dart Energy