

BAB 2

LANDASAN TEORI

2.1. Teori Basis Data

2.1.1. Pengertian Data

Menurut Atzeni et al. (2003, p2), data merupakan suatu bentuk penyimpanan informasi yang harus diterjemahkan terdahulu untuk menghasilkan suatu informasi. Data tidak mempunyai makna, tetapi setelah diterjemahkan dan dihubungkan dengan benar, data menghasilkan informasi yang memungkinkan kita meningkatkan pengetahuan.

Menurut Whitten et al. (2004, p23), data adalah fakta mentah mengenai orang, tempat, kejadian, dan hal-hal penting yang ada dalam organisasi. Tiap fakta, tanpa disertai fakta lainnya, secara relatif tidak mempunyai arti.

2.1.2. Pengertian Sistem

Sistem adalah sekelompok unsur yang berhubungan erat satu dengan lainnya, yang berfungsi untuk mencapai tujuan tertentu (Mulyadi, 1997, p2).

Menurut O'Brien (2002, p8), sistem merupakan sebuah kumpulan dari elemen-elemen yang saling berhubungan atau berinteraksi yang membentuk suatu kesatuan.

2.1.3. Pengertian Basis-Data dan Sistem Basis-Data

Menurut Connolly dan Begg (2005, p15), basis data merupakan suatu kumpulan data yang secara logika terhubung, dan dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Artinya basis-data adalah merupakan tempat penyimpanan data yang besar dan dapat digunakan secara bersamaan oleh banyak pemakai.

Menurut Date (2000, p5), basis-data adalah komputerisasi sistem penyimpanan data yang bertujuan untuk menyimpan dan memelihara informasi serta mengizinkan pengguna untuk mengambil kembali maupun memperbaharui informasi tersebut.

2.1.4. Pengertian Sistem Manajemen Basis-Data (DBMS)

Sistem Manajemen Basis-Data adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, menciptakan, memelihara, dan mengontrol akses ke dalam basis-data (Connolly dan Begg, 2005, p16).

Menurut Whitten et al. (2004, p554), sistem basis-data merupakan suatu perangkat lunak khusus yang digunakan untuk menciptakan, mengendalikan, serta mengelola sebuah basis-data.

Pada intinya setiap DBMS memiliki dua bagian utama, yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). Menurut Connolly dan Begg (2005, p40), *Data Definition Language* adalah bahasa yang memperbolehkan seorang *database administrator* (DBA) atau user untuk mendeskripsikan nama dari suatu entitas, atribut,

hubungan yang dibutuhkan oleh aplikasi bersamaan dengan integritas data dan keamanan datanya. Sedangkan untuk melakukan manipulasi data digunakan *Data Manipulation Language* merupakan bahasa yang memberikan fasilitas pengoperasian data yang ada dalam basis-data (Connolly, 2005, p41). Pengoperasian data yang akan dimanipulasi meliputi:

- a. penambahan data baru kedalam basis-data
- b. perubahan data yang telah tersimpan dalam basis-data
- c. pengambilan data yang terdapat dalam basis-data
- d. penghapusan data dari basis-data

Menurut Connolly dan Begg (2005, p16), secara umum, DBMS menyediakan beberapa fasilitas, yaitu:

1. mengizinkan pengguna untuk mendefinisikan basis-data menggunakan *Data Definition Language (DDL)*
2. mengizinkan pengguna untuk memasukkan, mengubah, menghapus data dari basis-data menggunakan *Data Manipulation Language (DML)*
3. DBMS memberikan akses yang terkontrol dalam basis-data, diantaranya yaitu:
 - a. Sistem keamanan, merupakan mekanisme yang menjaga hanya pengguna yang memiliki hak yang dapat mengakses basis-data
 - b. Sistem integritas, yang memelihara konsistensi data yang tersimpan dalam basis data dan meyakinkan perubahan data dalam basis-data mengikuti aturan

- c. *Concurrency control system*, merupakan mekanisme basis-data diubah secara benar pada saat banyak pengguna pada saat yang bersamaan mengubah data pada basis-data
- d. *Recovery Control System*, merupakan mekanisme untuk memperbaiki basis-data pada saat data mengalami kerusakan
- e. Sistem katalog yang dapat diakses oleh pengguna
- f. Independensi data, sebuah DBMS menyertakan fasilitas untuk mendukung program independen dari struktur aktual basis-data

2.1.5. Komponen Sistem Manajemen Basis Data

Menurut Connolly dan Begg (2005, p18), dalam suatu lingkungan Sistem Manajemen Basis-Data terdapat 5 komponen utama yaitu:

1. Perangkat Keras (*hardware*), digunakan oleh DBMS serta aplikasinya untuk beroperasi
2. Perangkat Lunak (*software*), merupakan komponen perangkat lunak yang terdiri dari DBMS dan program-program aplikasi, termasuk sistem operasi dan perangkat lunak jaringan apabila menggunakan jaringan komputer
3. Data, merupakan komponen paling penting dari suatu DBMS dilihat dari sudut pandang pengguna. Data memegang peranan sebagai penghubung antara komponen mesin dengan manusia dalam lingkungan DBMS
4. *Procedures*, merupakan instruksi serta aturan yang diterapkan untuk mendesain dan menggunakan basis-data

5. Manusia, merupakan komponen terakhir pada DBMS yang terlibat dalam sistem, terdiri atas *application programmer*, pengguna akhir (*end-users*), serta *database administrator*

2.1.6. Kelebihan dan Kekurangan Sistem Manajemen Basis-Data (DBMS)

Menurut Connolly dan Begg (2005, p26), kelebihan DBMS antara lain:

1. Kontrol terhadap redundansi data
2. Konsistensi data
3. Lebih banyak informasi dari jumlah yang sama
4. *Sharing data*
5. Meningkatkan integritas data
6. Meningkatkan keamanan
7. Penetapan standard
8. Lebih ekonomis
9. Menyeimbangkan kebutuhan yang bertentangan
10. Meningkatkan akses dan respon data
11. Meningkatkan produktivitas
12. Meningkatkan pemeliharaan melalui independensi data
13. Meningkatkan konkurensi data
14. Meningkatkan pelayanan *backup* dan *recovery*

Kekurangan basis-data antara lain:

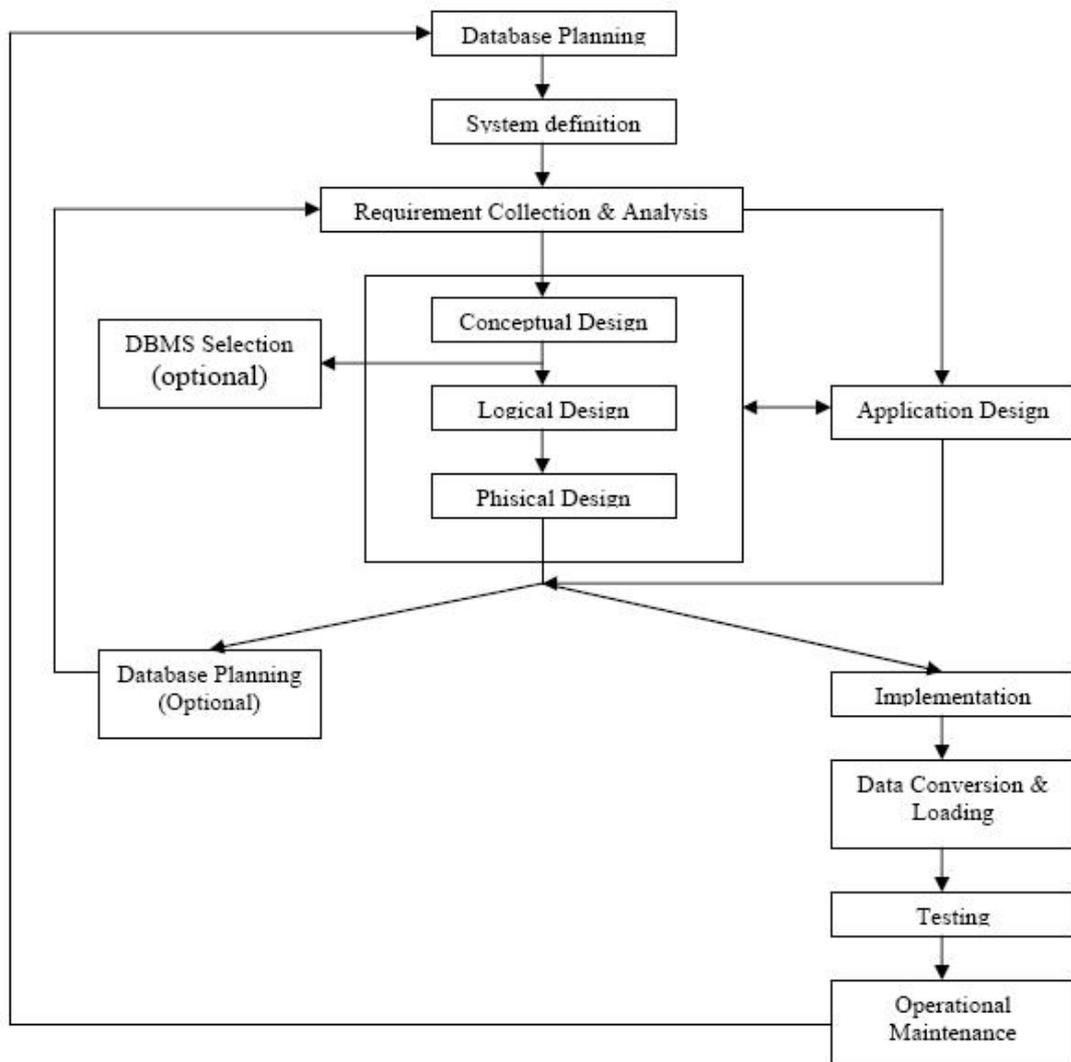
1. Kompleksitas

2. Ukuran
3. Biaya DBMS
4. Biaya penambahan perangkat keras
5. Biaya konversi
6. Performa
7. Dampak kegagalan yang lebih tinggi

2.1.7. Siklus Hidup Sistem Basis-Data

Untuk merancang aplikasi sistem basis-data diperlukan tahapan-tahapan terstruktur yang harus diikuti. Tahapan-tahapan terstruktur ini dinamakan Siklus Hidup Aplikasi Basis-Data (*Database Application Lifecycle*) atau DBLC.

Tahapan dari database application lifecycle menurut Connolly dan Begg dapat dilihat pada gambar 2.1



Gambar 2.1 Siklus Hidup Sistem Basis-Data

2.1.7.1. Perencanaan Basis-Data (*Database Planning*)

Perencanaan basis-data adalah aktivitas manajemen yang merencanakan tahapan dari siklus hidup basis-data agar dapat direalisasikan seefisien dan seefektif mungkin. Terdapat tiga

pertimbangan utama dalam menyusun sebuah strategi sistem informasi, yaitu:

1. Mengidentifikasi rencana dan sasaran (*goal*) perusahaan dengan mengikuti kebutuhan sistem informasi.
2. Mengevaluasi sistem informasi saat ini untuk menentukan kelebihan dan kekurangan.
3. Mempertimbangkan peluang IT yang dapat memberikan keuntungan kompetitif.

Tahapan dalam perencanaan basis-data juga harus menjelaskan:

1. *Mission statement* dari proyek basis-data. *Mission statement* menjelaskan tujuan utama aplikasi basis-data, membantu menjelaskan tujuan proyek basis-data, dan menyediakan maksud yang lebih jelas dalam pembuatan aplikasi basis-data secara efektif dan efisien (Connolly and Begg,2005,p286).
2. *Mission Objectives*. Selain merumuskan tujuan dari sebuah proyek basis-data, harus diperhatikan juga mengenai tugas apa saja yang harus didukung oleh basis-data tersebut. Setiap *mission objective* akan menjelaskan tugas tertentu yang harus didukung oleh basis-data, dengan asumsi jika basis-data mendukung mission objective, maka mission statement juga akan tercapai (Connolly and Begg,2005,p286)

2.1.7.2. Definisi Sistem (*System Definition*)

Definisi sistem menurut Connolly dan Begg (2005,p286) adalah mendeskripsikan runag lingkup dan batasan dari aplikasi sistem basis-data dan sudut pandang *user* yang dominan.

Sudut pandang *user* (*user view*) adalah mendefinisikan kebutuhan apa saja dari aplikasi basis-data dari perspektif tokoh yang spesifik (contohnya *Manager* atau *Supervisor*) atau area aplikasi perusahaan (seperti *marketing*, personalia, atau pengadaan barang) (Connolly and Begg,2005,p286).

2.1.7.3. Pengumpulan dan Analisis Kebutuhan (*Requirements Collection and Analysis*)

Dalam tahap ini, dilakukan proses pengumpulan dan analisis informasi tentang bagian organisasi yang akan didukung oleh aplikasi basis-data dan menggunakan informasi ini untuk mengidentifikasi kebutuhan pengguna terhadap sistem yang baru (Connolly and Begg, 2005, p288).

Ada tiga pendekatan untuk merancang kebutuhan aplikasi basis-data dengan banyak sudut pandang *user*, yaitu:

1. Pendekatan Terpusat (*Centralized Approach*)

Pendekatan terpusat menggabungkan setiap sudut pandang user ke dalam satu set kebutuhan untuk aplikasi basis-data yang baru.

2. Pendekatan Sudut Pandang Terintegrasi (*View Integration Approach*)

Pendekatan sudut pandang terintegrasi melibatkan setiap sudut pandang user untuk digunakan untuk membangun model data yang terpisah yang merepresentasikan sudut pandang itu. Hasil dari data model kemudian digabungkan pada saat mendesain basis-data.

3. Kombinasi dari Kedua Pendekatan

Suatu proses resmi menggunakan teknik-teknik seperti wawancara atau kuesioner untuk mengumpulkan fakta-fakta tentang sistem dan kebutuhan-kebutuhannya yang dinamakan dengan teknik fact-finding. Kegiatan yang digunakan dalam teknik ini yaitu:

- a. Memeriksa dokumen
- b. Wawancara
- c. Mengamati operasional perusahaan (observasi)
- d. Penelitian
- e. Studi kepustakaan

2.1.7.4. Desain Basis-Data (*Database Design*)

Desain basis-data adalah proses merancang desain basis-data yang akan mendukung tujuan dan operasional perusahaan (Connolly dan Begg, 2005, p291). Dalam merancang suatu basis-data, digunakan metodologi-metodologi yang membantu dalam tahap perencanaan basis-data. Metodologi perancangan (*Design Methodology*) adalah pendekatan struktur dengan menggunakan prosedur, teknik, alat serta bantuan dokumen untuk membantu dan memudahkan dalam proses perancangan. Teknik metodologi desain ini membantu dalam merencanakan, mengatur, mengontrol, dan mengevaluasi database development project (Connolly dan Begg, 2005, p438).

Proses dalam metodologi perancangan dibagi menjadi 3 tahap:

1. Perancangan Basis-Data Konseptual (*Conceptual Database Design*)
2. Perancangan Basis-Data Logikal (*Logical Database Design*)
3. Perancangan Basis-Data Fisikal (*Physical Database Design*)

Menurut Connolly dan Begg (2005, p292), ada dua pendekatan dalam perancangan basis-data, yaitu:

1. *Bottom-up*

Pendekatan ini dimulai dari tingkat paling dasar dari atribut (yakni properti dari entitas dan hubungan relasional) di mana melalui analisi gabungan antara atribut-atribut, dikelompokkan ke

dalam relasi-relasi yang merepresentasikan tipe-tipe entitas dan hubungan antar entitas.

2. *Top-down*

Pendekatan ini dimulai dari pengembangan model data yang terdiri dari beberapa hubungan relasional dan entitas tingkat tinggi.

Teori pendekatan perancangan basis-data yang digunakan pada skripsi ini menggunakan pendekatan *top-down*. Pendekatan ini berguna untuk mendesain basis-data yang kompleks dengan atribut yang banyak, di mana ditemukan kesulitan untuk membangun semua *functional dependencies* di antara atribut-atribut yang ada.

Top-down approach diilustrasikan menggunakan konsep dari model hubungan entitas (Entity Relationship Model), dimulai dengan mengidentifikasi entitas-entitas dan hubungan di antara entitas yang merupakan fokus penting dari perusahaan.

2.1.7.5. **Pemilihan DBMS (*DBMS Selection*)**

Pertimbangan dalam pemilihan DBMS dipengaruhi oleh faktor-faktor di bawah ini:

- a. Kemudahan Penggunaan (*Ease of Use*)
- b. Keandalan (*Reliability*)
- c. Biaya (*Cost*)
- d. Keamanan (*Security*)

- e. Kompatibel (*Compatibility*)
- f. Stabilitas Vendor (*Vendor Stability*)
- g. Pengembangan (*Development*)
- h. Kebutuhan Sistem (*System Requirement*)

2.1.7.6. Desain Aplikasi (*Application Design*)

Desain aplikasi yaitu rancangan tampilan pengguna dan program aplikasi yang menggunakan dan memproses basis-data.

Desain aplikasi dibagi menjadi dua aspek penting, yaitu:

a. Desain Transaksi (*Transaction Design*)

Menurut Connolly dan Begg (2005,p300), transaksi diartikan sebuah atau rangkaian aksi yang dilakukan oleh pengguna atau program aplikasi, yang mengakses atau mengubah isi basis-data.

Tipe utama transaksi yaitu:

1. *Retrieval Transaction*
2. *Update Transaction*
3. *Mixed Transaction*, gabungan dari *Retrieval Transaction* dan *Update Transaction*

b. Desain Antarmuka Pengguna (*User Interface Design*)

2.1.7.7. Prototyping

Prototyping adalah proses pembuatan model dari pekerjaan dari aplikasi basis-data. Tujuan utama mengembangkan *prototype* aplikasi basis-data adalah untuk memungkinkan *user* untuk mengidentifikasi fitur dari sistem yang berkerja baik, kurang lengkap atau jika memungkinkan, untuk memberikan saran untuk meningkatkan maupun memberikan fitur baru ke dalam aplikasi basis-data (Connolly dan Begg,2005,p304). Dua jenis strategi *prototype* yang umum digunakan yaitu:

- a. *Prototype* kebutuhan (*Requirement Prototyping*)
- b. *Prototype* evaluasi (*Evaluationary Prototyping*)

2.1.7.8. Implementasi (Implementation)

Implementasi adalah proses perealisasiian fisik dari desain basis-data dan desain aplikasi (Connolly dan Begg,2005,p304). Proses implementasi basis-data dijalankan menggunakan *Data Definition Language (DDL)* dari DBMS yang dipilih atau berdasarkan antarmuka pengguna, di mana memberikan fungsionalitas yang sama bersama dengan penyembunyian kalimat-kalimat DDL level rendah.

Program aplikasi diimplementasikan menggunakan bahasa pemrograman yang dipilih. Bagian dari program aplikasi ini adalah transaksi basis-data yang diimplementasikan menggunakan *Data Manipulation Language (DML)* dari DBMS terpilih.

2.1.7.9. Konversi dan Pengambilan Data (*Data Conversion and Loading*)

Konversi dan pengambilan data adalah proses pemindahan semua data yang ada ke dalam basis-data yang baru dan mengubah semua aplikasi yang ada untuk dijalankan dalam basis-data yang baru (Connolly dan Begg,2005,p305).

2.1.7.10. Pengujian (*Testing*)

Pengujian menurut Connolly dan Begg (2005,p305) adalah proses pengekseskuan program aplikasi dengan tujuan untuk mencari kesalahan.

2.1.7.11. Pemeliharaan Operasional (*Operational Maintenance*)

Pemeliharaan operasional menurut Connolly dan Begg (2005,p306) adalah proses monitoring dan pemeliharaan sistem setelah proses instalasi.

Aktivitas-aktivitas yang dilakukan dalam tahapan ini adalah:

- a. Memonitor performa sistem
- b. Memelihara dan meningkatkan aplikasi basis-data (jika diperlukan).

Proses monitoring berlanjut sepanjang umur dari aplikasi basis data dan dalam suatu waktu akan mengarah ke reorganisasi basis-data untuk memenuhi kebutuhan yang berubah.

2.1.8. Teori Metodologi Perancangan Basis-Data

Menurut Connolly dan Begg (2005, p438), Metodologi Perancangan adalah pendekatan terstruktur yang menggunakan prosedur, teknik, alat-alat, dan dokumentasi untuk memproses dan memfasilitasi proses perancangan.

Metodologi perancangan basis-data terbagi atas 3 bagian utama, yaitu:

2.1.8.1. Perancangan Basis-Data Konseptual

Perancangan basis-data konseptual merupakan proses membangun suatu model informasi yang digunakan di dalam perusahaan atau organisasi yang tidak tergantung dari semua pertimbangan fisik (Connolly, 2005, p439).

Langkah 1 : Membangun model lokal konseptual data untuk setiap *view*

Tujuannya adalah untuk memecah rancangan menjadi tugas-tugas yang lebih mudah untuk diatur dengan memeriksa sudut pandang pengguna yang beragam dari suatu perusahaan atau organisasi (Connolly, 2005, p442).

Langkah-langkah yang dilakukan dalam pembuatan model lokal konseptual data adalah sebagai berikut:

2.1.8.1.1. Mengidentifikasi Tipe Entitas

Bertujuan untuk mengidentifikasi tipe-tipe entitas utama yang dibutuhkan. Setelah teridentifikasi, entitas diberi nama

yang tepat dan jelas bagi pengguna. Nama dan penjelasan dari masing-masing entitas dimasukkan ke dalam kamus data.

Tabel Kamus Data

Nama Entitas	Deskripsi	Alias	Occurence

Tabel 2.1 Kamus Data

2.1.8.1.2. Mengidentifikasi Tipe Relasi

Tujuan dari tahapan ini adalah untuk mengidentifikasi relasi-relasi yang penting yang ada diantara tipe-tipe entitas yang telah terlebih dahulu diidentifikasi. Nama dari suatu relasi menggunakan kata kerja seperti mempunyai, dan lain-lain. Dalam tahap ini yang perlu diperhatikan adalah:

- Menggambarkan entitas-entitas yang telah diidentifikasi serta relasi dengan menggunakan diagram ER (*Entity Relationship*)
- Menentukan *multiplicity constraints* dari tiap-tiap relasi
- Memeriksa apakah terdapat fan traps dan chasm traps

Menurut Connolly dan Begg (2005, p364), fan traps adalah keadaan dimana suatu model merepresentasikan suatu relasi antar entitas tersebut menumbulkan kerancuan.

Chasm traps merupakan keadaan dimana suatu model sebaiknya memiliki sebuah relasi antar entitas, tetapi relasi itu tidak dibuat antar entitas tersebut (Connolly, 2005, p365)

- Memeriksa apakah tiap entitas memiliki paling tidak satu relasi dengan entitas lainnya
- Mendokumentasikan tipe relasi

2.1.8.1.3. Mengidentifikasi dan menghubungkan atribut suatu entitas

Langkah ini bertujuan untuk menghubungkan atribut-atribut dengan entitas-entitas atau relasi yang sesuai. Dalam langkah ini perlu diperhatikan hal-hal sebagai berikut:

- Menentukan apakah suatu atribut termasuk *simple* atau *composite attribut*. *Composite attribut* adalah suatu atribut yang terdiri dari lebih satu *simple attribut*
- Menentukan apakah suatu atribut termasuk dalam *single-valued* atau *multi valued*
- Mengidentifikasi *derived attribut*

Setelah atribut teridentifikasi, masing-masing atribut didokumentasikan yang memuat informasi sebagai berikut:

- Nama dan deskripsi atribut
- Tipe data dan ukuran field
- Alias atau nama lain dari atribut

- Apakah atribut tersebut *simple* atau *composite attribute*
- Apakah atribut tersebut termasuk *multi-valued*
- Apakah atribut termasuk *derived attribute*
- Nilai awal untuk atribut tersebut

2.1.8.1.4. Menentukan atribut domain

Pada tahap ini bertujuan untuk menentukan batasan suatu atribut dalam model data konseptual. Contohnya, menentukan atribut jenis kelamin pada entitas Karyawan, dengan 'L' atau 'P' atau nilai atribut idKaryawan yang ditentukan panjangnya 5 digit, dimana 2 digit awal berupa huruf dan 3 digit terakhir berupa angka dari 1-999.

2.1.8.1.5. Menentukan atribut kandidat dan primary key

Bertujuan untuk menentukan candidate key dari setiap entitas, apabila terdapat lebih dari satu candidate key maka tentukan salah satunya menjadi primary key. Candidate key merupakan atribut yang ditentukan secara minimal dari sebuah entitas yang secara unik dapat digunakan untuk mengidentifikasi setiap hubungan dari entitas tersebut.

2.1.8.1.6. Mempertimbangkan penggunaan *Enhanced Modelling Concept*

Tahap ini bertujuan untuk mempertimbangkan apakah akan menggunakan konsep pemodelan yang ditingkatkan, seperti spesialisasi, generalisasi, spesialisasi, dan komposit. Spesialisasi dan generalisasi merupakan proses pengelompokan beberapa entitas menghasilkan entitas baru. Spesialisasi menggunakan analisis *top-down* dan generalisasi menggunakan *bottom-up*.

2.1.8.1.7. Mengecek redundansi

Bertujuan untuk memeriksa redundansi data maupun informasi dalam model. Langkah-langkah yang ditempuh untuk melakukan pengecekan, yaitu:

- Memeriksa kembali hubungan *one-to-one* (1:1)
- Menghilangkan relasi yang redundan

2.1.8.1.8. Memvalidasi model konseptual lokal terhadap transaksi-transaksi pengguna

Tahap bertujuan untuk memastikan bahwa model konseptual mendukung transaksi-transaksi yang dibutuhkan oleh *view*. Ada dua pendekatan yang dapat dilakukan untuk menjamin bahwa model lokal konseptual telah mendukung transaksi tersebut, yaitu:

- Mendeskripsikan transaksi
- Menggunakan alur transaksi

2.1.8.1.9. Meninjau ulang data konseptual lokal dengan pengguna

Tujuan dari langkah ini adalah untuk menjamin model lokal konseptual data telah merepresentasikan *view* dengan benar antara pengguna dengan model.

2.1.8.2. Perancangan Basis-Data Logikal

Menurut Connolly dan Begg (2005, p439), Perancangan Basis-Data Logikal merupakan proses membangun suatu model informasi yang digunakan oleh suatu organisasi maupun perusahaan berdasarkan model data spesifik, tetapi tidak bergantung dari DBMS (*Database Management System*) khusus dan pertimbangan fisikal lainnya.

Langkah 2 : Membangun dan memvalidasi model lokal logikal data

Bertujuan untuk menerjemahkan model data konseptual ke dalam sebuah logikal data model. Kemudian memvalidasi model untuk menjamin struktur yang benar dan mendukung kebutuhan transaksi (Connolly, 2005, p462).

Pada perancangan basis-data logikal langkah kedua, langkah-langkah yang ditempuh antara lain:

2.1.8.2.1. Menurunkan relasi-relasi untuk model data logikal

Bertujuan untuk membuat relasi bagi model data logikal untuk mewakili entitas, relasi, dan atribut yang telah diidentifikasi. Setiap relasi dideskripsikan dengan menggunakan *Database Defenition Language (DBDL)*. Dengan menggunakan DBDL, langkah awal yang dilakukan adalah melakukan spesifikasi nama dari relasi yang diikuti dengan daftar dari relasi atribut yang mudah. Kemudian mengidentifikasikan *primary key* dan *foreign key* dari suatu relasi.

Dalam menentukan tempat dari atribut *foreign key*, terlebih dahulu entitas *parent* dan *child* yang terlibat dalam relasi diidentifikasi. Entitas *parent* mengarah menjadi entitas yang menjadi tempat *primary key*, kemudian *primary key* di salin ke relasi yang mewakili entitas anak, sehingga *primary key* dalam entitas induk bertindak sebagai *foreign key*.

2.1.8.2.2. Memvalidasikan relasi dengan menggunakan normalisasi

Tujuannya adalah untuk melakukan validasi terhadap relasi yang terdapat dalam model lokal logikal data menggunakan teknik-teknik normalisasi. Normalisasi menjamin model yang dihasilkan mendekati model yang mewakili suatu organisasi atau perusahaan, mempunyai struktur yang konsisten, redundansi minimal dan stabilitas yang maksimal.

2.1.8.2.3. Memvalidasikan relasi terhadap transaksi pengguna

Bertujuan untuk menjamin relasi dalam model lokal logikal data mendukung transaksi yang dibutuhkan oleh *view*.

2.1.8.2.4. Mengecek *integrity constraints*

Integrity constraint merupakan batasan yang digunakan untuk menjaga agar basis-data menjadi konsisten.

Ada 5 tipe integrity constraints, yaitu:

- *Required data* (data yang valid)
- Batasan *domain atribut*
- *Entity integrity* (*primary key* tidak boleh null)
- *Referential integrity* (*foreign key* pada suatu entitas harus sesuai dengan *candidate key* dari entitas lain)
- *Enterprise constraint* (batasan pada organisasi)

2.1.8.2.5. Meninjau ulang model data logikal dengan pengguna

Tahapan ini bertujuan untuk menjamin model data logikal dan dukungan dari dokumentasi yang berisi penjelasan dari model telah mewakili *view* dengan tepat.

2.1.8.2.6. Menggabungkan model data logikal menjadi model global (langkah opsional)

Pada tahap ini tujuan yang ingin dicapai adalah menggabungkan model data logikal menjadi sebuah model data global yang merepresentasikan *view* pengguna terhadap database. Setiap model data logikal menghasilkan suatu Diagram ER, skema relasi, kamus data, dan dokumentasi pendukung yang menjelaskan model. Pada langkah ini, seluruh komponen digunakan untuk mengidentifikasi kemiripan dan perbedaan antar model dan membantu menyatukannya ke dalam sebuah model data global.

Tahap-tahap yang dilakukan antara lain:

1. menggabungkan model data logikal lokal ke dalam model global
2. melakukan validasi model data logikal global
3. meninjau kembali logikal data global

2.1.8.2.7. Memeriksa pertumbuhan yang akan datang

Bertujuan untuk menentukan apakah ada perubahan yang signifikan seperti keadaan yang tidak terduga di masa yang akan datang dan untuk melakukan penilaian apakah model data global logikal dapat menyesuaikan dengan perubahan ini.

2.1.8.3. Perancangan Basis-Data Fisikal

Menurut Connolly dan Begg (2005, p496), Perancangan Basis-Data fisikal adalah proses untuk menghasilkan gambaran implementasi

dari basis-data di tempat penyimpanan sekunder. Menjelaskan relasi dasar, organisasi *file*, dan indeks yang digunakan untuk mengakses data secara efisien, serta *integrity constraints* yang berkaitan dengan faktor keamanan.

Langkah 3 : Menerjemahkan model data logikal untuk DBMS yang dituju

Bertujuan untuk menghasilkan suatu skema basis-data relasional dari model data logikal yang dapat dimplementasikan ke dalam DBMS yang dituju.

Pada langkah ini terdapat tiga langkah yang akan dilakukan yaitu:

2.1.8.3.1. Merancang relasi dasar

Bertujuan untuk menentukan bagaimana merepresentasikan dasar relasi yang teridentifikasi dalam model data logikal pada DBMS yang dituju. Untuk memulai proses perancangan fisika, langkah awal yang dilakukan adalah mengumpulkan dan memahami informasi mengenai relasi yang dihasilkan selama perancangan basis-data logikal. Informasi yang dibutuhkan diperoleh dari kamus data dan definisi dari relasi dijelaskan dengan menggunakan *Database Design Language (DBDL)*. Untuk setiap relasi yang teridentifikasi dalam model data logikal, diperoleh penjelasan dari relasi meliputi:

- a. Nama dari relasi
- b. Daftar dari *simple attributes* dalam *brackets*

- c. *Primary key* (PK) dan letaknya yang tepat, *alternate keys* (AK) dan *foreign keys* (FK)
- d. Daftar dari setiap *derived attributes* dan bagaimana menghitungnya
- e. *Referential integrity* constraints untuk setiap *foreign keys* yang teridentifikasi

Dari kamus data dapat diperoleh penjelasan dari setiap atribut:

- a. *Domain*, terdiri dari tipe data, panjang, dan setiap *constraints* dari *domain*
- b. Pilihan nilai *default* untuk atribut
- c. Apakah atribut dapat memiliki nilai Null

Langkah selanjutnya adalah mengimplementasikan relasi dasar. Keputusan untuk mengimplementasikannya tergantung dari DBMS yang dituju. Perancangan dari relasi dasar harus didokumentasikan seluruhnya bersama dengan alasan kenapa memilih rancangan yang diajukan.

2.1.8.3.2. Merancang representasi dari *derived data*

Bertujuan untuk menentukan bagaimana merepresentasikan setiap *derived data* yang ada dalam model data logikal pada DBMS yang dituju. Atribut yang nilainya didapatkan dengan cara memeriksa nilai dari atribut yang lain disebut dengan

derived atau *calculated attributes*. Dibawah ini adalah contoh dari *derived data*, yaitu:

- a. Total gaji bulanan untuk seluruh karyawan
- b. Jumlah karyawan yang bekerja di satu cabang

Perancangan dari *derived data* harus didokumentasikan seluruhnya dengan alasan kenapa memilih desain yang diajukan.

2.1.8.3.3. Merancang *general constraints*

Bertujuan untuk merancang *general constraints* untuk DBMS yang dituju. Untuk merancangnya, tergantung kepada DBMS yang akan digunakan. Perancangan *general constraints* harus didokumentasikan secara khusus alasan kenapa memilih salah satu dimana terdapat banyak pilihan.

Langkah 4 : Mendesain organisasi *file* dan indeks

Bertujuan untuk menentukan pengorganisasian file yang optimal untuk mendapatkan relasi dasar dan indeks yang dibutuhkan untuk mendapatkan performa yang baik.

Pada perancangan basis-data fisik langkah keempat, langkah-langkah yang dilakukan adalah:

2.1.8.3.4. Menganalisis transaksi

Bertujuan untuk memahami fungsi-fungsi dari transaksi yang akan dijalankan di basis-data dan untuk menganalisis transaksi

yang penting. Kriteria performa yang harus diidentifikasi dalam menganalisis transaksi adalah:

1. transaksi yang sering dijalankan dan akan berdampak signifikan terhadap performa
2. transaksi yang kritis terhadap operasi bisnis
3. waktu selama sehari/seminggu ketika ada kebutuhan tinggi yang ada dalam basis-data (*peak load*)

2.1.8.3.5. Memilih organisasi file

Bertujuan untuk menentukan pengorganisasian file yang efektif untuk setiap relasi dasar. Ada beberapa cara untuk pengorganisasian file tergantung dari jenis file tersebut. Cara-cara tersebut antara lain *Heap*, *Hash*, *Indexed sequential access method (ISAM)*, *B-tree*, *Clusters*

2.1.8.3.6. Memilih indeks

Untuk menentukan apakah penambahan indeks akan meningkatkan performa dari sistem. Pendekatan yang digunakan untuk memilih pengorganisasian file yang tepat untuk suatu relasi adalah dengan mempertahankan tabel tidak terurut susunannya dan membuat sebanyak *secondary indexes* diperlukan. Oleh karenanya, pilih atribut untuk *ordering* atau *clustering* tabel, yaitu:

1. Merupakan atribut yang sering digunakan untuk join operations untuk membuat lebih efisien
2. atribut yang sering digunakan untuk mengakses relasi di dalam susunan atribut

2.1.8.3.7. Memperkirakan kapasitas disk yang diperlukan

Langkah 5 : Merancang *view* pengguna

Bertujuan untuk merancang pandangan yang telah teridentifikasi selama pengumpulan dan analisis kebutuhan dalam siklus hidup aplikasi relasional basis-data.

Contohnya, Staff yang terdiri dari dua view, yaitu view untuk Supervisor dan view untuk Asisten

Langkah 6 : Merancang mekanisme keamanan

Membatasi pengaksesan basis-data kepada pengguna yang tidak mempunyai hak, dan melakukan spesifikasi pengguna terhadap basis-data yang dapat diakses. Keamanan Sistem menjangkau akses dan penggunaan basis data d sistem level seperti username dan password. Kemananan Data menjangkau akses dan penggunaan objek basis-data dan hak akses pengguna terhadap objek dalam basis-data

Langkah 7 : Mempertimbangkan pengenalan dan redundansi kontrol

Pada langkah ini, mempertimbangkan *denormalisasi* skema rasional untuk meningkatkan performa. Hasil dari normalisasi adalah rancangan basis-data logikal yang secara struktural konsisten dan mengurangi jumlah redundansi. Faktor yang menjadi pertimbangan, yaitu:

1. Denormalisasi membuat implementasi lebih kompleks
2. Denormalisasi mengorbankan fleksibilitas
3. Denormalisasi meningkatkan performa dalam *retrieve data* namun lambat dalam *updates*

Ukuran performa dari suatu perancangan basis-data dapat dilihat dari sudut pandang tertentu yaitu melalui pendekatan efisiensi data (normalisasi) atau pendekatan efisiensi proses (denormalisasi). Efisiensi data dimaksudkan untuk meminimalkan kapasitas disk, dan efisiensi proses bertujuan untuk mempercepat proses saat mengambil data dari basis-data.

2.1.9. *Entity Relationship Modelling (ER Modelling)*

Model ER merupakan pendekatan top-down dalam perancangan basis-data yang dimulai dengan mengidentifikasi data-data penting (*entity*) dan hubungan (*relationship*) antara data-data tersebut dan merepresentasikannya dalam sebuah model (Connolly, 2002, p330).

2.1.9.1. Entity Type

Menurut Connolly dan Begg (2002, p331), entitas merupakan kumpulan objek atau kejadian yang memiliki sifat-sifat yang sama, diidentifikasi oleh sebuah organisasi, dan memiliki sifat yang independen. Entitas dilambangkan dengan kotak dengan nama entitasnya dalam kotak tersebut dalam Model ER. Setiap objek diidentifikasi secara unik disebut *entity occurrences*.

Entitas dapat dibagi menjadi 2 bagian, yaitu:

- *Strong Entity Type*
- *Weak Entity Type*

2.1.9.2. Relationship Type

Menurut Connolly dan Begg (2002, p334), tipe relasi merupakan suatu set asosiasi yang memiliki arti diantara tipe-tipe entitas. Tipe relasi mempunyai nama yang mendeskripsikan fungsinya. Dalam Model ER, tipe relasi dilambangkan dengan garis yang menghubungkan tipe entitas yang berasosiasi dan diberi label sesuai dengan nama relasi.

- Derajat Tipe Relasi

Merupakan jumlah partisipan pada tipe relasi. Partisipan adalah entitas yang terlibat dalam suatu tipe relasi. Relasi dengan derajat dua disebut *binary*. Relasi dengan derajat tiga

disebut *ternary*. Relasi dengan derajat empat disebut *quaternary*.

- Relasi Rekursif

Merupakan tipe relasi dimana entitas yang sama berpartisipasi lebih dari satu kali dengan peranan yang berbeda

2.1.9.3. Atribut

Atribut merupakan properti tertentu dari sebuah entitas dan tipe relasi (Connolly, 2002, p331). Atribut dapat dikelompokkan sebagai berikut:

2.1.9.3.1. Atribut Sederhana dan Komposit

Atribut sederhana merupakan atribut yang terdiri dari satu komponen tunggal dan keberadaannya bersifat independen. Atribut sederhana tidak dapat dipecah menjadi atribut yang lebih kecil lagi.

Atribut komposit adalah atribut yang dibentuk dari banyak komponen, dimana setiap komponennya bersifat independen.

2.1.9.3.2. Atribut *Single-valued* dan *multi-valued*

Atribut *single-valued* merupakan atribut yang menyimpan nilai tunggal untuk setiap kejadian tipe entitas. Umumnya atribut merupakan *single-valued*.

Atribut *multi-valued* adalah atribut yang menyimpan lebih dari satu nilai untuk setiap kejadian tipe entitas.

2.1.9.3.3. Atribut Turunan

Atribut turunan merupakan atribut yang dimana nilainya diperoleh ataupun diturunkan dari nilai atribut atau sekelompok atribut yang sama.

2.1.9.3.4. Keys

Jenis-jenis *keys* yang digunakan dalam Model ER, yaitu:

a. *Candidate Key*

Merupakan sekumpulan minimal atribut yang secara unik digunakan untuk mengidentifikasi setiap kejadian dari tipe entitas

b. *Composite Key*

Merupakan dua atau lebih atribut yang membentuk sebuah candidate key. Untuk mengidentifikasi sebuah tipe entitas, nilai dari kedua atribut tersebut harus digabungkan.

c. *Primary Key*

Merupakan candidate key yang terpilih secara unik untuk mengidentifikasi setiap kejadian dari tipe entitas

d. *Alternate Key*

Merupakan candidate key yang tidak terpilih sebagai primary key

e. *Foreign Key*

Merupakan atribut key yang melengkapi sebuah hubungan yang mereferensikan ke induknya

2.1.9.4. Batasan Struktural

Batasan struktural menggambarkan hubungan yang sesuai dengan kenyataan (real-world) pada setiap entitas yang ikut dalam hubungan dan dinyatakan dengan *multiplicity*. Menurut Connolly dan Begg (2002, p344), *multiplicity* merupakan jumlah *occurance* yang mungkin terjadi dari tipe entitas yang berhubungan kepada sebuah *occurance* dari tipe entitas lain pada suatu hubungan.

Derajat yang digunakan pada suatu hubungan yaitu *binary relationship* yang terdiri atas:

- a. *One-to-One (1:1) Relationship*
- b. *One-to-Many (1:*) Relationship*
- c. *Many-to-Many (*:*) Relationship*

2.1.9.5. Permasalahan dengan Model

Dalam membangun model ER, terdapat beberapa masalah yang mungkin timbul diakibatkan oleh kesalahan interpretasi dari makna suatu relasi tertentu. Biasanya disebut dengan *connection traps*. Connection traps dapat dibagi menjadi 2 tipe yaitu, *fan traps* dan *chasm traps*.

2.1.10. Normalisasi

2.1.10.1. Pengertian Normalisasi

Menurut Connolly dan Begg (2005, p376), normalisasi merupakan teknik yang digunakan untuk menghasilkan kumpulan atau set relasi dari properti-properti yang dikehendaki, memberikan kebutuhan data dari sebuah organisasi.

Tujuan dari normalisasi adalah untuk menjamin struktur yang konsisten, kerangkapan yang minimal, serta stabilitas struktur data yang maksimal.

Manfaat yang dapat diperoleh dari proses normalisasi, yaitu:

- a. Jumlah kapasitas penyimpanan data menjadi minimal
- b. Mengurangi resiko ketidakkonsistenan data dalam suatu basis-data
- c. Menimalkan kemungkinan *update* dan *delete anomaly*
- d. Memaksimalkan stabilitas dari struktur data

2.1.10.2. Proses Normalisasi

Proses normalisasi terkait dengan serangkaian aturan yang digunakan dalam memeriksa sebuah relasi, sehingga sebuah basis-data dapat dinormalisasi dalam beberapa tahap. Normalisasi dilaksanakan dalam serangkaian langkah secara bertahap dengan tujuan membuat relasi semakin kuat. Semakin tinggi langkah yang dikenakan maka akan semakin kuat relasi yang terbentuk.

Secara umum, langkah-langkah dalam melakukan normalisasi dapat dibagi kedalam 3 level atau tingkatan normalisasi, yaitu Bentuk Normal

Pertama (1NF), Bentuk Normal Kedua (2NF), dan Bentuk Normal Ketiga (3NF)

2.1.10.2.1. Bentuk Normal Pertama (1NF)

Dalam tahap Bentuk Normal Pertama (1NF), kita melakukan identifikasi dan membuang atribut yang berulang (*repeating group*) dan atribut yang memiliki nilai lebih dari satu. Syarat suatu hubungan yang memenuhi bentuk normal pertama adalah:

- a. Setiap dari baris dan kolom berisi atribut yang bernilai tunggal
- b. *Primary Key* telah ditentukan
- c. Setiap atribut yang bernilai ganda (*multi value*) telah dihilangkan

2.1.10.2.2. Bentuk Normal Kedua (2NF)

Bentuk Normal Kedua (2NF) merupakan sebuah relasi yang terdapat dalam 1NF yang dimana setiap atribut bukan *primary key* bersifat *full function dependency* terhadap *primary key* dari relasi tersebut (Connolly, 2002, p392).

Menurut Connolly dan Begg (2002, p391), dalam suatu relasi $A \rightarrow B$, A dan B merupakan atribut dari relasi tersebut. Maka, B termasuk dalam *full functional dependency* jika B secara fungsional bergantung pada A, tetapi B bukanlah anggota himpunan A.

Syarat suatu hubungan termasuk bentuk normal kedua, yaitu:

- a. Telah memenuhi bentuk normal pertama (1NF)
- b. Bebas terhadap atribut yang bergantung sebagian. Setiap atribut yang bergantung sebagian (*partial independen*) dari suatu table telah dipindahkan ke sebuah tabel baru yang terpisah
- c. Setiap atribut dalam tabel bergantung penuh secara fungsional terhadap primary key.

2.1.10.2.3. Bentuk Normal Ketiga (3NF)

Bentuk Normal Ketiga (3NF) merupakan sebuah relasi yang memenuhi bentuk normal pertama dan bentuk normal kedua dimana tidak terdapat atribut non-primary key yang bersifat *transitively independent* dari primary key-nya (Connolly, 2002, p394).

Menurut Connolly dan Begg (2002, p394), dalam sebuah relasi $A \rightarrow B$ dan $B \rightarrow C$, A, B, C merupakan atribut-atribut relasi tersebut, Maka, C dikatakan *transitively independent* terhadap A melalui B. A tidak bergantung secara fungsional terhadap B maupun C.

Syarat suatu hubungan termasuk dalam bentuk normal ketiga, yaitu:

- a. Telah memenuhi bentuk normal pertama (1NF) dan kedua (2NF)
- b. Bebas terhadap atribut yang bergantung secara transitif. Setiap atribut yang bergantung secara transitif dipindahkan dalam sebuah tabel baru yang terpisah

2.1.11. *Structured Query Language (SQL)*

2.1.11.1. **Pengertian SQL**

Menurut Kadir (2002, p101), Structured Query Language (SQL) merupakan bahasa query standar yang digunakan untuk mengakses basis-data relasional.

2.1.11.2. **Kelompok Statemen dalam SQL**

2.1.11.2.1. **Data Definition Language (DDL)**

Menurut Connolly dan Begg (2005, p40), *Data Definition Language* adalah bahasa yang memperbolehkan seorang *database administrator (DBA)* atau user untuk mendeskripsikan nama dari suatu entitas, atribut, hubungan yang dibutuhkan oleh aplikasi bersamaan dengan integritas data dan keamanan datanya.

DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-attribut basis-data, *table*, kolom, batasan-batasan terhadap suatu atribut serta hubungan antar table.

Statemen DDL antara lain *CREATE TABLE*, *ALTER TABLE*, *DROP TABLE*, *CREATE INDEX*, dan *DROP INDEX*.

2.1.11.2.2. Data Manipulation Language (DML)

Data Manipulation Language merupakan bahasa yang memberikan fasilitas pengoperasian data yang ada dalam basis-data (Connolly, 2005, p41). Pengoperasian manipulasi data diantaranya memasukkan data baru, mengubah data, maupun menghapus data dari basis-data.

Terdapat 2 tipe DML menurut Connolly (2005, p41-42), yaitu:

a. *Procedural DML*

DML prosedural merupakan bahasa yang mengizinkan pengguna (programmer) dalam memberikan instruksi kepada sistem mengenai data apa yang dibutuhkan dan bagaimana cara pengambilan datanya. Dengan kata lain, pengguna harus menjelaskan operasi pengaksesan data dengan menggunakan prosedur yang ada untuk mendapatkan informasi yang dibutuhkan.

b. *Non-Procedural DML*

DML non-prosedural merupakan bahasa yang memungkinkan pengguna untuk menentukan data apa yang dibutuhkan dengan menyatakan spesifikasinya tanpa menjelaskan bagaimana prosedur yang harus dilaksanakan untuk mendapatkannya.

2.1.11.2.3. Data Control Language (DCL)

Dalam DCL berisi perintah-perintah untuk mengendalikan pengaksesan data. Pengendalian dapat dilakukan pada setiap pengguna, table, kolom, ataupun operasi yang boleh dilakukan. Perintah-perintah ini berkaitan dengan manajemen keamanan basis-data. Perintah-perintah yang termasuk diantaranya *GRANT*, *REVOKE*, dan *LOCK TABLE*.

2.1.11.2.4. Pengendali Transaksi

Pengendali transaksi merupakan perintah-perintah yang berfungsi untuk mengendalikan pengeksekusian suatu transaksi. Perintah yang termasuk dalam DCL diantaranya *COMMIT*, dan *ROLLBACK*

2.1.11.2.5. Pengendali Programatik

Dalam kelompok statemen ini, perintah-perintah yang termasuk dalam pengendali programatik merupakan pernyataan-pernyataan yang berhubungan dengan pemanfaatan SQL dalam bahasa lain. Pernyataan-pernyataan terdapat pada bahasa konvensional generasi ketiga (3-GL), seperti COBOL. Perintah-perintah yang termasuk dalam kategori ini diantaranya *CLOSE*, *DECLARE*, *FETCH*, dan *OPEN*.

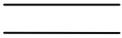
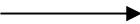
2.2. Teori-teori lain

2.2.1. *Data Flow Diagram (DFD)*

Menurut (McGraw-Hill, 2004, p357), Data Flow Diagram adalah aliran data antara sistem dengan lingkungannya, atau di antara 2 proses di dalam sistem. Aliran data yang direpresentasikan adalah suatu *input* ke dalam proses atau *output* data dari suatu proses. Aliran data juga merepresentasikan pembuatan, pembacaan, penghapusan, dan pengubahan data pada suatu *file* atau basis-data.

McLeod (2001, p401) menyatakan, Data Flow Diagram merupakan representasi grafis dari sistem yang menggunakan beberapa simbol untuk mengilustrasikan bagaimana aliran data melewati proses-proses yang saling berhubungan.

Simbol DFD secara umum, yaitu:

1. Terminal, menunjukkan sumber dan tempat tiba data 
2. Proses, pelaksanaan tugas atau fungsi 
3. Penyimpanan data 
4. Aliran data, saluran komunikasi 

DFD dapat dibagi menjadi 3 bagian yaitu:

1. Diagram hubungan, menggambarkan hubungan yang sedang dianalisis dengan eksternal entitasnya
2. Diagram nol, menggambarkan proses-proses penting yang ada pada sistem yang dianalisis dan yang akan dibuat

3. Diagram rinci, menggambarkan proses-proses yang lebih rinci dan mendetail mengenai sistem yang dianalisis dan yang akan dirancang

2.2.2. *Entity Relationship Diagram (ERD)*

Menurut Fathansyah (1999, p79), *Entity Relationship Diagram (ERD)* menggambarkan komponen-komponen dari entitas dan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari keadaan sesungguhnya (real-world) yang akan kita tinjau.

ER diagram juga dapat menyatakan keseluruhan struktural logika dari basis-data menggunakan bagan (Silberschatz, 2002, p42).

2.3. Teori Pendukung

2.3.1. Pengertian Puskesmas

Menurut Trihono dalam buku “Arrimes Manajemen Puskesmas Berbasis Paradigma Sehat”, pengertian puskesmas adalah unit pelaksana teknis dinas kesehatan kabupaten/kota yang bertanggung jawab menyelenggarakan pembangunan kesehatan di suatu wilayah kerja. Dengan rincian masing-masing sebagai berikut :

1. Unit Pelaksana Teknis

Sebagai unit pelaksana teknis (UPTD) dinas kesehatan kabupaten/kota, puskesmas berperan menyelenggarakan sebagian dari tugas teknis operasional dinas kesehatan kabupaten/kota dan merupakan unit pelaksanaan tingkat pertama serta ujung tombak pembangunan kesehatan di Indonesia.

2. Pembangunan Kesehatan

Pembangunan kesehatan adalah penyelenggaraan upaya kesehatan oleh bangsa Indonesia untuk meningkatkan kesadaran, kemauan, dan kemampuan hidup sehat bagi setiap orang agar terwujud derajat kesehatan masyarakat yang optimal.

3. Pertanggungjawaban Penyelenggaraan

Penanggungjawab utama penyelenggaraan seluruh upaya pembangunan kesehatan di wilayah kabupaten/kota adalah dinas kesehatan kabupaten/kota, sedangkan puskesmas bertanggung jawab hanya untuk sebagian upaya pembangunan kesehatan yang

dibebankan oleh dinas kesehatan kabupaten/kota sesuai dengan kemampuannya.

4. Wilayah kerja

Secara nasional, standar wilayah kerja puskesmas adalah satu kecamatan. Tetapi apabila di satu kecamatan terdapat lebih dari satu puskesmas, maka tanggung jawab wilayah kerja dibagi antar puskesmas, dengan memperhatikan keutuhan konsep wilayah (desa/kelurahan atau RW). Masing-masing puskesmas tersebut secara operasional bertanggungjawab langsung kepada dinas kesehatan kabupaten/kota.

2.3.2. Fungsi dan Tujuan Puskesmas

Ada tiga fungsi puskesmas menurut Trihono (Arrimes Manajemen Puskesmas Berbasis Paradigma Sehat,2005), yaitu :

1. Pusat penggerak pembangunan berwawasan kesehatan

Puskesmas selalu berupaya menggerakkan dan memantau penyelenggaraan pembangunan lintas sector termasuk oleh masyarakat dan dunia usaha di wilayah kerjanya, sehingga berwawasan serta mendukung pembangunan kesehatan.

Di samping itu puskesmas aktif memantau dan melaporkan dampak kesehatan dari penyelenggaraan setiap program pembangunan di wilayah kerjanya. Khusus untuk pembangunan kesehatan, upaya yang dilakukan puskesmas adalah mengutamakan

pemeliharaan kesehatan dan pencegahan penyakit tanpa mengabaikan penyembuhan penyakit dan pemulihan kesehatan.

2. Pusat pemberdayaan masyarakat

Puskesmas selalu berupaya agar perorangan terutama pemuka masyarakat, keluarga dan masyarakat termasuk dunia usaha memiliki kesadaran, kemauan dan kemampuan melayani diri sendiri dan masyarakat untuk hidup sehat, berperan aktif adalah memperjuangkan kepentingan kesehatan termasuk sumber pembiayaannya, serta ikut menetapkan, menyelenggarakan dan memantau pelaksanaan program kesehatan. Pemberdayaan perorangan, keluarga, dan masyarakat ini diselenggarakan dengan memperhatikan kondisi dan situasi, khususnya social budaya masyarakat setempat.

3. Pusat pelayanan kesehatan strata pertama

Puskesmas bertanggungjawab menyelenggarakan pelayanan kesehatan tingkat pertama secara menyeluruh, terpadu dan berkesinambungan. Pelayanan kesehatan tingkat pertama yang menjadi tanggungjawab puskesmas meliputi :

a. Pelayanan kesehatan perorangan

Pelayanan kesehatan perorangan adalah pelayanan yang bersifat pribadi (private goods) dengan tujuan utama menyembuhkan penyakit dan pemulihan kesehatan perorangan, tanpa mengabaikan pemeliharaan kesehatan dan pencegahan penyakit. Pelayanan perorangan tersebut adalah

rawat jalan dan untuk puskesmas tertentu ditambah dengan rawat inap.

b. Pelayanan kesehatan masyarakat

Pelayanan kesehatan masyarakat adalah pelayanan yang bersifat public (public goods) dengan tujuan utama memelihara dan meningkatkan kesehatan serta mencegah penyakit tanpa mengabaikan penyembuhan penyakit dan pemulihan kesehatan. Pelayanan kesehatan masyarakat tersebut antara lain adalah promosi kesehatan, pemberantasan penyakit, penyehatan lingkungan, perbaikan gizi, peningkatan kesehatan keluarga, keluarga berencana, kesehatan jiwa masyarakat serta berbagai program kesehatan masyarakat lainnya.

Tujuan pembangunan kesehatan yang diselenggarakan oleh puskesmas adalah mendukung tercapainya tujuan pembangunan kesehatan nasional, yakni meningkatkan kesadaran, kemauan dan kemampuan hidup sehat bagi setiap orang yang bertempat tinggal di wilayah kerja puskesmas, agar terwujud derajat kesehatan yang setinggi-tingginya dalam rangka mewujudkan Indonesia Sehat 2010.

2.3.3. Jasa Pelayanan di Puskesmas

Menurut dokumen Profil Kesehatan kab. Maros Tahun 2006, pelayanan medis yang disediakan oleh puskesmas, mencakup:

1. Penyelenggaraan pelayanan kesehatan dasar
 - a. Pelayanan kesehatan ibu dan bayi
 - b. Pelayanan kesehatan anak pra-sekolah dan usia sekolah
 - c. Pelayanan keluarga berencana (KB)
 - d. Pelayanan imunisasi
 - e. Pelayanan pengobatan/perawatan:
 - Cakupan rawat jalan
 - Cakupan rawat inap
 - f. Pelayanan gangguan jiwa
2. Penyelenggaraan perbaikan gizi masyarakat
 - a. Pemantauan pertumbuhan balita
 - b. Pelayanan gizi
 - c. Pelayanan obstetrik dan neonatal emergensi dasar dan komprehensif
3. Penyelenggaraan pelayanan kesehatan rujukan dan penunjang
 - a. Pelayanan gawat darurat
4. Penyelenggaraan pemberantasan penyakit menular
 - a. Penyelenggaraan, penyidikan epidemiologi dan penanggulangan KLB dan gizi buruk
 - b. Pencegahan dan pemberantasan penyakit folio

- c. Pencegahan dan pemberantasan penyakit TB paru
 - d. Pencegahan dan pemberantasan penyakit ISPA
 - e. Pencegahan dan pemberantasan penyakit HIV/AIDS
 - f. Pencegahan dan pemberantasan penyakit DBD
 - g. Pencegahan dan pemberantasan penyakit diare
5. Penyelenggaraan kesehatan lingkungan dan sanitasi dasar
- a. Pelayanan kesehatan lingkungan
 - b. Pelayanan pengendalian vektor
 - c. Pelayanan kesling, hygiene sanitasi di tempat umum
6. Penyelenggaraan promosi kesehatan
- a. Penyuluhan perilaku sehat
7. Pencegahan dan penanggulangan penyalahgunaan narkotika, psikotropika dan zat adiktif (P3-NAPZA)
- a. Penyuluhan P3-NAPZA yang berbasis masyarakat
8. Penyelenggaraan pelayanan kefarmasian (obat)
- a. Pelayanan penyediaan obat dan perbekalan kesehatan
 - b. Pelayanan penggunaan obat generik
9. Penyediaan pembiayaan dan jaminan kesehatan
- a. Penyelenggaraan pembiayaan untuk gakin dan masyarakat rentan
 - b. Jenis pelayanan yang dilaksanakan sesuai kebutuhan:
 - Pelayanan kesehatan kerja
 - Pelayanan kesehatan usia lanjut
 - Pelayanan gizi

- Pencegahan dan pemberantasan penyakit malaria
- Pencegahan dan pemberantasan penyakit kusta
- Pencegahan dan pemberantasan penyakit filariasis
- Sarana air bersih dan toilet keluarga

2.3.4. Rekam Medis

Rekam medis adalah pencatatan informasi pasien yang berguna untuk menyimpan data pasien serta riwayat penyakit yang pernah diderita dan pengobatan yang pernah dijalani.

Basbeth (*Basbeth, Ferryal, 2005, rekam medis, <http://www.freewebs.com/medicalrecord>*) menyajikan kumpulan definisi rekam medis dalam berbagai keputakaan.

Definisi rekam medis:

1. Menurut Permenkes No. 749a/Menkes/Per/XII/1989

Rekam medis adalah berkas yang berisi catatan dan dokumen mengenai identitas pasien, hasil pemeriksaan, pengobatan, tindakan dan pelayanan lainnya yang diterima pasien pada sarana kesehatan, baik rawat jalan, maupun rawat inap.

2. Menurut Edna K Huffman

Rekam medis adalah berkas yang menyatakan siapa, apa, mengapa, di mana, kapan, dan bagaimana pelayanan yang diperoleh seorang pasien selama dirawat atau menjalani pengobatan.

3. Menurut Gemala Hatta

Rekam medis merupakan kumpulan fakta tentang kehidupan seseorang dan riwayat penyakitnya, termasuk keadaan sakit, pengobatan saat ini dan masa lampau yang ditulis oleh para praktisi kesehatan dalam upaya mereka memberikan pelayanan kesehatan kepada pasien.

2.3.5. Kebutuhan Komputerisasi Medis

Beberapa masalah yg dikutip oleh Aluise (1997, p167) dari pernyataan Sellars dalam pedoman untuk para dokter dan karyawan kantor yang mengidentifikasi kebutuhan komputer dalam praktik medis antara lain sebagai berikut:

1. Pertumbuhan praktik

Praktik medis sedang berkembang menjadi organisasi skala besar sewaktu ditambahkan pelayanan kesehatan yang baru dan memperluas fungsi-fungsi klinis dan bisnis. Kenaikan jumlah pasien mengakibatkan kenaikan dalam jumlah karyawan dan pekerjaan pencatatan sehingga membutuhkan rutinitas kerja yang lebih efisien. Komputer mungkin tidak akan menambah bisnis baru, tetapi dapat membantu dalam mengatur dan mengelola pertumbuhan.

2. Jadwal pengajuan penagihan tidak tercapai

Memproses laporan keuangan bulanan pasien dan klaim asuransi merupakan suatu tugas yang menyita waktu serta membutuhkan

ketelitian serta ketepatan waktu. Hal ini dapat menjadi masalah kalau tuntutan praktik lainnya harus diprioritaskan. Penggunaan komputer untuk perhitungan dan pengajuan tagihan dalam melakukan sebagian besar pekerjaan tanpa keterlibatan staf.

3. Keputusan berdasarkan informasi

Tenaga profesional pelayanan kesehatan memerlukan sejumlah besar informasi guna mengambil keputusan di bidang bisnis dan medis. Banyak informasi pada sistem manual yang tidak mudah diakses pada saat dibutuhkan. Sistem terkomputerisasi memungkinkan informasi tentang pasien dapat diberikan secara rinci dan cepat.

4. Pembatasan fasilitas dan ruang

Sistem komputer mungkin memakan tempat yang lebih sedikit dibandingkan dengan dokumen-dokumen pada sistem manual.