

BAB 2

LANDASAN TEORI

2.1 Teori-teori Basis Data

Berikut ini akan dijelaskan tentang teori-teori basis data, yaitu mulai dari pengertian sistem basis data, siklus hidup aplikasi basis data sampai pada tahapan perancangan basis data.

2.1.1 Sistem Basis Data

Berikut ini akan dijelaskan pengertian sistem basis data, dimulai dari pengertian sistem, pengertian basis data, sampai pada pengertian sistem basis data itu sendiri.

2.1.1.1 Pengertian Sistem

Menurut Fathansyah (2002, p9) sistem adalah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi atau tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu.

Menurut Kamus Umum Bahasa Indonesia (Poerwadarminta 1991, p955) sistem diartikan menjadi tiga, yaitu :

1. Sekelompok bagian-bagian (alat) yang bekerja bersama-sama untuk melakukan sesuatu maksud.
2. Sekelompok dari pendapat, peristiwa kepercayaan yang disusun dan diatur baik-baik.
3. Cara (metode) yang teratur untuk melakukan sesuatu.

Dari definisi diatas, dapat disimpulkan bahwa sistem adalah sekelompok bagian-bagian yang saling berhubungan dan bekerja bersama-sama untuk melakukan sesuatu.

2.1.1.2 Pengertian Basis Data

Menurut Connolly & Begg (2002, p14) basis data adalah kumpulan data yang secara logika berhubungan, dan merupakan gambaran dari data tersebut, dan dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.

Menurut Fathansyah (2002, p2) basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

Menurut Kristanto (1994, p3) basis data merupakan kumpulan file-file yang mempunyai kaitan antara satu file dengan yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.

Dari definisi diatas, dapat disimpulkan bahwa basis data adalah kumpulan data yang saling berhubungan dan disimpan secara bersama untuk memenuhi kebutuhan informasi dari suatu organisasi.

2.1.1.3 Pengertian Sistem Basis Data

Menurut Silberschatz (1986, p1) sistem basis data dirancang untuk mengatur sekumpulan informasi. Pengaturan data meliputi ketentuan struktur untuk penyimpanan informasi dan ketetapan mekanisme untuk manipulasi

informasi. Sistem basis data harus menyediakan informasi yang aman untuk disimpan dalam basis data, meskipun sistem rusak ataupun usaha akses bagi yang tidak memiliki hak.

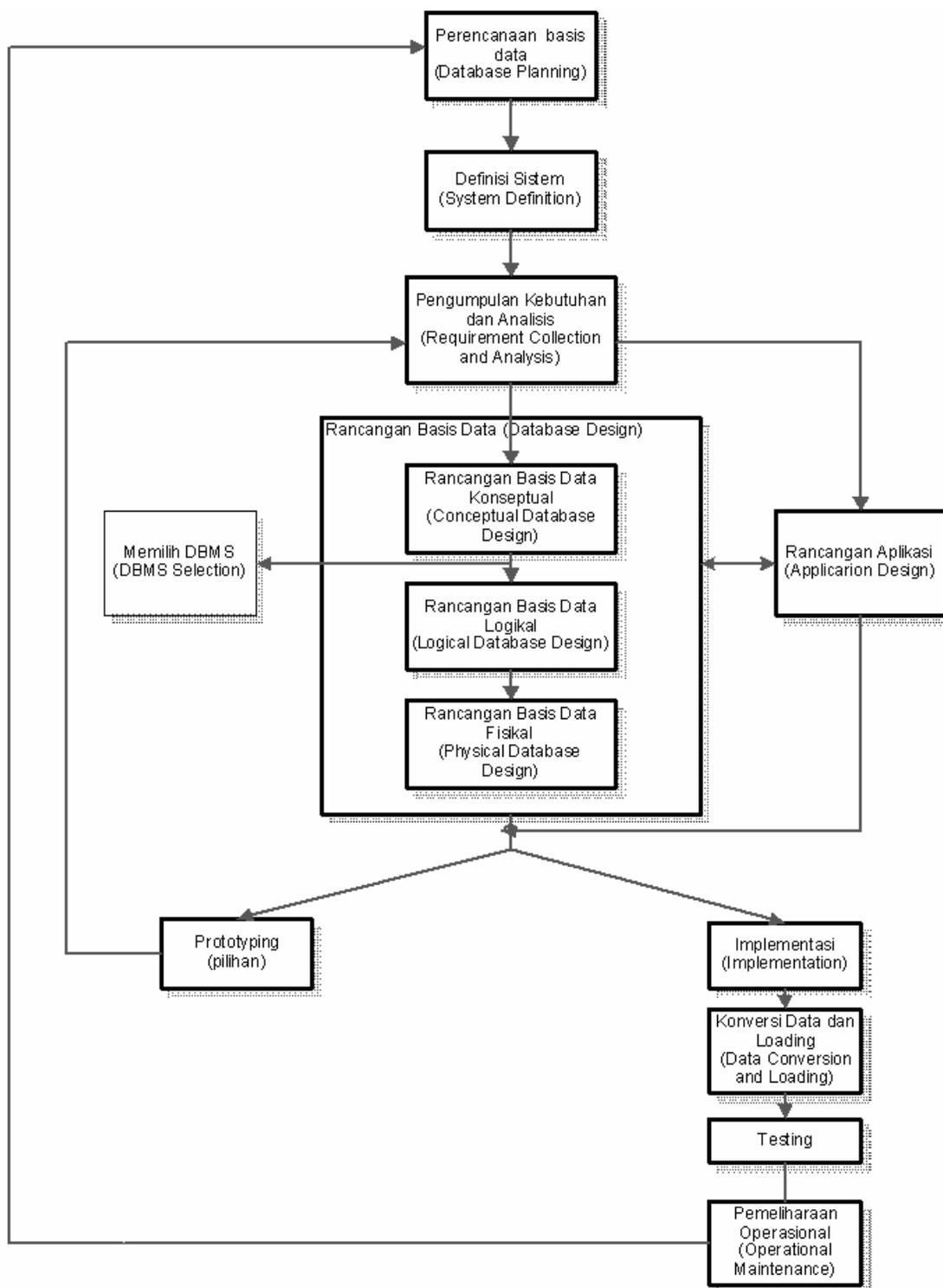
Menurut Connolly & Begg (2002, p4) sistem basis data merupakan kumpulan dari aplikasi-aplikasi program yang berinteraksi dengan basis data.

Menurut Fathansyah (2002, p9) sistem basis data adalah sistem yang terdiri atas kumpulan file (tabel) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi file-file (tabel-tabel) tersebut.

Dari definisi diatas, dapat disimpulkan bahwa sistem basis data adalah kumpulan aplikasi program yang berinteraksi dengan basis data dan memungkinkan pemakai dan atau program lain untuk mengakses dan memanipulasinya serta menyediakan informasi yang aman untuk disimpan dalam basis data.

2.1.2 Siklus Hidup Aplikasi Basis Data (Database Application Lifecycle)

Tahap-tahap siklus hidup aplikasi basis data menurut Connolly and Begg (2002, p270) adalah:



Gambar 2.1 Siklus Hidup Aplikasi Basis Data

Sumber: Connolly & Begg(2002), p272

2.1.2.1 Perencanaan basis data (*Database planning*)

Kegiatan utama dari perencanaan basis data adalah merencanakan bagaimana tahapan dalam siklus hidup dapat diwujudkan seefisien dan seefektif mungkin. Langkah pertama dalam perencanaan basis data adalah menentukan pernyataan misi untuk proyek basis data. Pernyataan misi mendefinisikan tujuan dari aplikasi basis data. Langkah selanjutnya adalah menentukan tujuan misi. Tiap tujuan misi harus menentukan tugas yang harus ada didalam basis data.

Perencanaan basis data juga meliputi perkembangan yang mengatur bagaimana data akan dikumpulkan, bagaimana format harus dikhususkan, dokumentasi apa yang dibutuhkan, dan bagaimana rancangan dan implementasi akan dijalankan.

2.1.2.2 Definisi sistem (*System Definition*)

Menurut Connolly & Begg (2002, p274) kegiatan utama definisi sistem adalah menentukan ruang lingkup dan batasan dari aplikasi basis data dan pandangan utama pemakai. Sebelum kita mencoba untuk merancang aplikasi basis data, pertama-tama kita harus menentukan ruang lingkup sistem yang akan kita selidiki. Sangatlah penting bagi kita untuk tidak hanya memasukkan pemakai yang sekarang dan lingkungan aplikasi, tapi juga perlu memasukkan pemakai mendatang dan aplikasinya kedalam ruang lingkup sistem.

2.1.2.3 Pengumpulan kebutuhan dan analisis (*Requirement collection and analysis*)

Pengumpulan kebutuhan dan analisis merupakan proses mengumpulkan dan menganalisis informasi tentang bagian dari perusahaan yang harus didukung oleh aplikasi basis data, dan menggunakan informasi ini untuk mengidentifikasi kebutuhan pemakai dari sistem yang baru.

Ada beberapa teknik dalam mengumpulkan informasi untuk perusahaan, salah satu diantaranya adalah teknik *fact-finding*. Informasi yang diperoleh untuk setiap pandangan pemakai meliputi:

- Gambaran tentang data yang digunakan atau yang dihasilkan.
- Keterangan tentang bagaimana data digunakan atau dihasilkan.
- Kebutuhan tambahan yang diperlukan untuk aplikasi basis data yang baru.

Informasi ini kemudian akan dianalisis untuk mengidentifikasi kebutuhan yang akan dimasukkan ke dalam aplikasi yang baru.

Ada tiga pendekatan untuk mengatur kebutuhan aplikasi basis data untuk pemakai yang banyak, yaitu:

- Pendekatan terpusat

Kebutuhan untuk setiap pemakai digabung kedalam sekumpulan kebutuhan untuk aplikasi basis data yang baru.

- Pendekatan penggabungan pandangan

Kebutuhan untuk setiap pemakai digunakan untuk membangun data model yang terpisah untuk menggambarkan pandangan pemakai itu sendiri. Data model yang dihasilkan kemudian akan digabung didalam tahap perancangan basis data.

- Campuran dari keduanya

2.1.2.4 Perancangan Basis Data (*Database design*)

Berdasarkan Connolly & Begg (2002, p279) perancangan basis data merupakan proses pembuatan sebuah rancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan. Ada dua pendekatan utama dari perancangan basis data:

- Pendekatan Bottom-up

Cocok untuk rancangan basis data yang sederhana dengan sejumlah kecil atribut.

Akan tetapi, pendekatan ini menjadi sulit ketika diterapkan kedalam rancangan basis data yang rumit dengan sejumlah besar atribut, dimana akan sulit untuk menentukan ketergantungan fungsional antara atribut-atribut tersebut.

- Pendekatan Top-down

Merupakan pendekatan yang cocok untuk rancangan basis data yang rumit.

Pendekatan ini dimulai dengan perkembangan data model yang mengandung beberapa entity dan hubungan dan kemudian menerapkan top-down untuk menentukan entity dan hubungan tingkat yang lebih rendah. Pendekatan ini digambarkan dengan menggunakan konsep dari ER(*Entity Relationship*) model.

Ada tiga tahap dalam perancangan basis data yaitu rancangan basis data konseptual (*Conceptual database design*), rancangan basis data logikal (*Logical database design*), rancangan basis data fisikal (*Physical database design*)

2.1.2.5 Memilih DBMS (*DBMS Selection*)

Pada tahap ini, memilih DBMS yang cocok untuk mendukung aplikasi basis data. Pendekatan sederhana untuk memilih DBMS adalah untuk mengecek fitur DBMS berdasarkan kebutuhan. Dalam memilih produk DBMS yang baru, ada kesempatan untuk memastikan bahwa proses pemilihan telah direncanakan dengan baik, dan sistem tersebut mengirim keuntungan pada perusahaan. Berikut merupakan langkah untuk memilih DBMS :

1. Tentukan waktu referensi belajar

Waktu referensi untuk memilih DBMS dibangun, menyatakan tujuan dan ruang lingkup dari belajar, dan tugas yang perlu dilakukan. Dokumen ini juga harus meliputi gambaran kriteria yang digunakan untuk mengevaluasi produk DBMS, dan semua batasan yang dibutuhkan serta waktu untuk belajar.

2. Buat daftar dua atau tiga produk

Kriteria yang dianggap penting untuk implementasi yang berhasil, dapat digunakan untuk menghasilkan daftar awal produk DBMS yang digunakan untuk evaluasi. Contoh, keputusan untuk memasukkan produk DBMS akan tergantung pada budget yang tersedia, tingkat dukungan vendor, kompatibilitas dengan software yang lain. Informasi tambahan produk dapat diperoleh dengan menghubungi pemakai yang sudah ada yang akan menyediakan keterangan tentang seberapa bagusnya dukungan vendor, tentang bagaimana produk mendukung aplikasi tertentu. Berdasarkan pemahaman tentang fungsi dan fitur produk DBMS, maka daftar pendek dari dua atau tiga produk akan ditentukan.

3. Evaluasi Produk

Ada berbagai fitur yang dapat digunakan untuk mengevaluasi produk DBMS. Untuk tujuan evaluasi, fitur ini dapat dinilai secara kelompok (contoh, definisi data) atau secara individu (contoh, jenis data yang tersedia).

4. Merekomendasikan pilihan dan menghasilkan laporan

Tahap akhir dari memilih DBMS adalah mendokumentasikan proses dan menyediakan pernyataan penemuan dan rekomendasi untuk produk DBMS tertentu.

2.1.2.6 Rancangan aplikasi (*Application design*)

Menurut Connolly & Begg (2002, p287), rancangan aplikasi merupakan rancangan antarmuka pemakai dan program aplikasi yang menggunakan dan memproses basis data. Dalam banyak kasus, tidak mustahil untuk melengkapi rancangan aplikasi sampai rancangan basis data itu sendiri mengambil tempat. Disisi lain, basis data ada untuk mendukung aplikasi, sehingga harus ada arus informasi antara rancangan aplikasi dan rancangan basis data.

Kita harus memastikan bahwa fungsi yang dinyatakan pada spesifikasi kebutuhan pemakai berada di dalam rancangan aplikasi untuk aplikasi basis data. Hal ini melibatkan perancangan program aplikasi yang mengakses basis data dan perancangan transaksi. Sebagai tambahan untuk merancang bagaimana fungsi yang diperlukan diperoleh, kita harus merancang antarmuka pemakai yang cocok untuk aplikasi basis data.

2.1.2.7 Prototyping

Menurut Connolly & Begg (2002, p291) Prototyping merupakan proses membangun model kerja dari aplikasi basis data. Prototype merupakan model kerja yang secara normal tidak mempunyai semua fitur yang dibutuhkan atau menyediakan semua fungsi dari sistem akhir. Tujuan dari pengembangan prototype aplikasi basis data adalah untuk memperbolehkan pemakai menggunakan prototype dalam mengenali fitur sistem yang bekerja dengan baik, dan jika mungkin untuk menyarankan perbaikan atau bahkan fitur baru bagi aplikasi basis data. Ada dua strategi prototyping yang umum digunakan saat ini:

- *Requirement prototyping* menggunakan prototype untuk menentukan kebutuhan aplikasi basis data yang diusulkan dan sekali kebutuhan telah lengkap, prototype dibuang.
- *Evolutionary prototyping* digunakan untuk tujuan yang sama, perbedaannya adalah bahwa prototype tidak dibuang tapi dengan pengembangan selanjutnya akan menjadi aplikasi basis data yang berjalan.

2.1.2.8 Implementasi (*Implementation*)

Menurut Connolly & Begg (2002, p292) Implementasi merupakan perwujudan fisik dari basis data dan rancangan aplikasi. Implementasi basis data diperoleh menggunakan DDL dari DBMS yang dipilih atau GUI, yang menyediakan fungsi sama ketika menyembunyikan pernyataan DDL tingkat rendah. Pernyataan DDL digunakan untuk membuat struktur basis data dan file basis data kosong.

Program aplikasi diterapkan dengan menggunakan generasi bahasa ketiga atau keempat. Bagian dari program aplikasi ini adalah transaksi basis data, yang diterapkan dengan menggunakan DML dari sasaran DBMS.

Keamanan dan pengaturan penyatuan untuk aplikasi juga diimplementasikan. Beberapa pengaturan ini diimplementasikan menggunakan DDL, tapi yang lain akan didefinisikan diluar penggunaan DDL.

2.1.2.9 Konversi data dan loading (*Data conversion and loading*)

Menurut Connolly & Begg (2002, p292) Konversi data dan loading mengirim data yang ada ke dalam basis data baru dan mengubah aplikasi yang ada untuk dijalankan pada aplikasi basis data yang baru. Tahap ini hanya diperlukan ketika sistem aplikasi basis data baru menggantikan sistem yang lama. Sekarang ini, sangatlah umum bagi DBMS untuk mempunyai perangkat yang mengisi file yang ada kedalam basis data baru. Perangkat biasanya memerlukan spesifikasi dari sumber file dan basis data sasaran, dan kemudian secara otomatis mengubah data ke format yang diperlukan dari file basis data yang baru.

2.1.2.10 Testing

Menurut Connolly & Begg (2002, p293) Testing merupakan proses menjalankan program aplikasi dengan maksud untuk menemukan kesalahan. Sebelum dijalankan, aplikasi basis data yang baru harus diuji. Hal ini diperoleh dengan menggunakan strategi uji yang direncanakan dengan hati-hati dan data nyata sehingga keseluruhan proses testing dapat dijalankan.

Pada kenyataannya, testing tidak dapat menunjukkan kehadiran kesalahan; hanya dapat menunjukkan bahwa kesalahan software ada. Sebagai keuntungan tambahan, testing menunjukkan bahwa basis data dan program aplikasi nampak berjalan menurut spesifikasi mereka dan kebutuhan penampilan mereka nampak terpenuhi. Setelah testing selesai, sistem aplikasi siap untuk diberikan kepada pemakai.

2.1.2.11 Pemeliharaan operasional (*Operational maintenance*)

Menurut Connolly & Begg (2002, p293) Pemeliharaan operasional adalah proses pengawasan dan menjaga sistem mengikuti instalasi. Pada tahap sebelumnya, aplikasi basis data telah diimplementasikan dan diuji.

2.1.3 Tahapan dalam perancangan basis data

Dalam merancang suatu basis data, ada tiga tahap yang harus dilakukan yaitu rancangan konseptual, rancangan logikal dan rancangan fisik. Berikut ini merupakan penjelasan singkat tentang langkah-langkah dari tahap dalam merancang basis data.

2.1.3.1 Rancangan basis data konseptual (*Conceptual database design*)

Rancangan basis data konseptual merupakan proses membangun model informasi yang digunakan dalam perusahaan, bebas dari semua pertimbangan fisik. Tahap ini melibatkan pembuatan model konseptual. Data model dibangun dengan menggunakan informasi yang didapatkan dalam spesifikasi kebutuhan pemakai. Langkah-langkah dalam membangun data model konseptual lokal untuk tiap pandangan adalah:

1. Identifikasi tipe entitas
2. Identifikasi tipe relasional
3. Identifikasi dan Asosiasi atribut dengan entitas atau relasional.
4. Tentukan domain atribut
5. Tentukan atribut dari kandidat dan *primary key*
6. Pertimbangkan penggunaan konsep *enhanced modeling* (pilihan)
7. Periksa model dari redundansi
8. Validasi model konseptual lokal berdasarkan transaksi pemakai
9. Ulangi data model konseptual lokal dengan pemakai.

2.1.3.2 Rancangan basis data logikal (*Logical database design*)

Menurut Connolly & Begg (2002, p281) rancangan basis data logikal merupakan proses membangun model informasi yang digunakan dalam perusahaan berdasarkan pada data model khusus, tapi bebas dari DBMS tertentu dan pertimbangan fisik lainnya. Data model konseptual yang kita dapatkan dari tahap sebelumnya, diperbaiki dan dipetakan kedalam data model logikal.

Menurut Kristanto (1994, p45) data model logikal adalah satu teknik untuk menjelaskan dengan baik struktur informasi bisnis dan aturan-aturan sebagai masukan pada proses perancangan basis data.

Kriteria untuk mendapatkan data model logikal yang paling optimal:

- Kebenaran struktur. Konsistensi dengan jalur definisi bisnis dan informasi organisasi.

- Kemudahan. Mudah untuk dimengerti oleh orang yang tidak punya pendidikan khusus atau pemakai sistem yang bukan profesional di komputer.
- Tidak redundansi. Tidak mempunyai informasi yang berlebihan yang ditulis berkali-kali, usahakan untuk satu potong informasi hanya disimpan benar-benar di satu tempat, tidak di beberapa tempat.
- Dapat dipakai bersama (*shareability*). Kondisi data sangat tidak spesifik dan hanya dapat dibaca oleh satu dua aplikasi atau teknologi, namun diharapkan dapat diakses oleh beberapa aplikasi dan teknologi.
- Mudah dikembangkan (*extensibility*). Mampu untuk menerima kebutuhan baru dengan akibat yang ringan terhadap perubahan data dasarnya.
- Kesatuan (*integrity*). Konsistensi dengan jalur bisnis yang digunakan dan pengelolaan informasi.

Dalam setiap bagian dari proses pengembangan data model logikal, model akan diuji dan divalidasi berdasarkan kebutuhan pemakai. Teknik normalisasi digunakan untuk menguji kebenaran dari data model logikal. Normalisasi menjamin bahwa hubungan yang diturunkan dari data model tidak akan menunjukkan redundansi, yang dapat menyebabkan *update anomalies* ketika diimplementasikan. Data model logikal merupakan sumber informasi untuk tahap selanjutnya, yaitu rancangan basis data fisik.

Langkah-langkah dalam membangun dan validasi data model lokal logikal untuk tiap pandangan:

1. Pindahkan fitur yang tidak kompatibel dengan model relasional (pilihan).

2. Turunkan relasi untuk data model logikal lokal.
3. Validasi relasi menggunakan normalisasi.
4. Validasi relasi berdasarkan transaksi pemakai.
5. Definisikan batasan integritas.
6. Ulangi data model logikal lokal dengan pemakai.

Langkah-langkah dalam membangun dan validasi data model logikal global:

1. Gabung data model logikal kedalam model global.
2. Validasi data model logikal global.
3. Periksa untuk pertumbuhan selanjutnya.
4. Ulangi data model logikal global dengan pemakai.

2.1.3.2.1 Normalisasi

Menurut Fathansyah (2002, p39) normalisasi merupakan pendekatan dalam membangun rancangan basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal.

Menurut Connolly & Begg (2002, p376) normalisasi adalah teknik untuk menghasilkan sekumpulan hubungan (*relation*) dengan properti yang dibutuhkan, dimana akan diberikan permintaan data-data dari perusahaan.

Normalisasi seringkali dijalankan sebagai serangkaian dari langkah-langkah. Beberapa langkah berhubungan dengan kondisi normal yang diketahui propertinya.

- *Unnormalized Form* (UNF)

Merupakan tabel yang terdiri dari satu atau lebih kelompok berulang (*repeating group*)

- *First Normal Form (1NF)*

Merupakan relasi dimana ada beberapa baris (*row*) dan kolom (*column*) yang mengandung satu atau hanya satu nilai saja tanpa mengandung data kelompok berulang (*repeating group*). Pada tahap ini, proses normalisasi dimulai dengan memindahkan data dari sumbernya ke dalam format tabel. Pada format ini, tabel adalah *unnormalized form* (UNF) dan menunjuk ke *unnormalized table*.

Untuk memindahkan UNF ke 1NF, terlebih dahulu mengidentifikasi dan memindahkan kelompok yang berulang dalam tabel. Kelompok berulang adalah atribut, kelompok dari atribut, dalam tabel yang terdapat nilai banyak untuk satu kejadian yang menunjuk atribut kunci untuk tabel tersebut. Pada 1NF telah ditentukan kunci utama (*primary key*) yang unik dan tidak pernah kosong, juga menghilangkan atribut yang memiliki operasi perhitungan.

- *Second Normal Form (2NF)*

Merupakan relasi yang ada di dalam *first normal form* (1NF) dan setiap atribut yang bukan kunci utama (*primary key*) secara penuh merupakan ketergantungan fungsional pada kunci utama. Pada tahap 2NF ini, telah dipisahkan ketergantungan parsial menjadi tabel-tabel tersendiri dengan kunci utama adalah atribut determinan masing-masing. Atribut determinan tetap ada pada tabel 1NF dan menjadi kunci referensi (*foreign*

key) bagi tabel baru. Atribut bukan kunci tergantung fungsional (*functional dependency*) pada atribut kunci.

- *Third Normal Form (3NF)*

Merupakan relasi yang ada di dalam *first normal form (1NF)* dan *second normal form (2NF)* dimana tidak ada atribut yang bukan kunci utama ketergantungan transitif (*transitive dependency*) pada kunci utama. Ketergantungan transitif telah dipisahkan menjadi tabel-tabel tersendiri. Pada tahap 3NF dapat digambarkan dalam *Entity – Relationship Diagram (ERD)* sebagai representasi gabungan dari seluruh dokumen atau struktur data yang ada didalam sistem.

2.1.3.3 Rancangan basis data fisikal (*Physical database design*)

Rancangan basis data fisikal merupakan proses menghasilkan gambaran implementasi dari basis data pada *secondary storage*; menggambarkan hubungan dasar, file perusahaan, dan index yang digunakan untuk memperoleh akses yang efisien ke data, dan *integrity constraint* serta keamanan.

Tahap ini merupakan tahap akhir dalam proses perancangan basis data, menjelang bagaimana perancang menentukan bagaimana basis data diimplementasikan. Dalam mengembangkan rancangan basis data fisikal, kita harus menentukan sasaran sistem basis data. Oleh karena itu, rancangan fisikal dihubungkan ke sistem DBMS.

Langkah-langkah dalam merancang basis data secara fisik adalah:

1. Perancangan relasi dasar
2. Analisis Transaksi

3. Pembuatan index untuk setiap entitas
4. Mengestimasi kapasitas penyimpanan yang dibutuhkan
5. Perancangan mekanisme keamanan

2.1.4 ERM (Entity relationship modelling)

Menurut Connolly & Begg (2002, p330) ERM adalah pendekatan top-down untuk merancang basis data yang dimulai dengan mengidentifikasi data-data penting yang disebut *entities* dan *relationship* diantara data yang harus direpresentasikan di dalam model.

2.1.4.1 Entitas (*entity*)

Merupakan suatu obyek yang diidentifikasi dalam lingkungan pemakai atau suatu obyek nyata yang dapat dibedakan dari sesuatu yang lain dan tidak saling tergantung. Entity digambarkan menggunakan persegi panjang (). Entity (entitas) dibedakan menjadi dua tipe, yaitu:

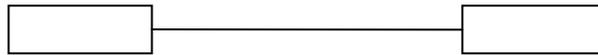
- *Strong Entity Type* (SET), yaitu tipe entitas yang tidak bergantung pada tipe entitas lainnya.
- *Weak Entity Type* (WET), yaitu tipe entitas yang bergantung pada tipe entitas lainnya.

2.1.4.2 Relasi (*relationship*)

Merupakan hubungan diantara sejumlah entitas yang ada. Relationship disimbolkan dalam bentuk intan/diamonds (). Terdapat beberapa tipe

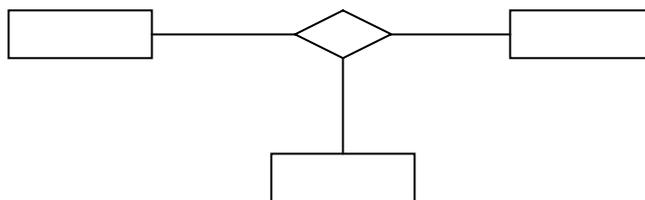
relationship jika dilihat berdasarkan derajat relationship (jumlah entitas yang berpartisipasi dalam suatu relationship) yaitu:

- Binary : relationship dengan derajat dua



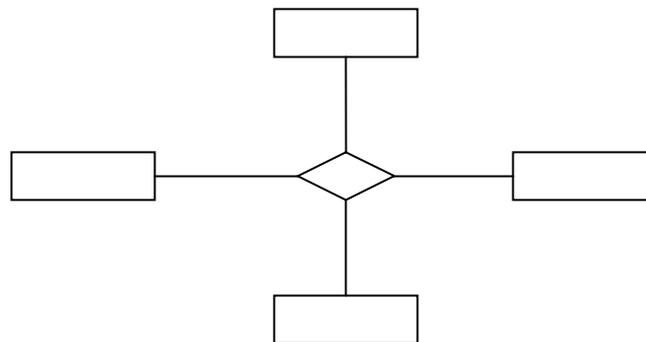
Gambar 2.2 relationship derajat dua

- Ternary : relationship dengan derajat tiga



Gambar 2.3 relationship derajat tiga

- Quaternary : relationship dengan derajat empat



Gambar 2.4 relationship derajat empat

Dikenal juga istilah *recursive relationship* yaitu tipe hubungan dimana entitas yang sama mengikutsertakan lebih dari satu peran yang berbeda (*different roles*). Recursive relationship terkadang juga disebut *unary relationship*.

2.1.4.3 Atribut (*attribute*)

Atribut berfungsi untuk menjelaskan entitas dan setiap atribut mempunyai nilai (harga). Atribut digambarkan dengan simbol ellips (). Setiap entity mempunyai atribut untuk mewakili suatu entity. Atribut dibedakan menjadi beberapa macam, yaitu:

- Atribut sederhana (*simple attribute*), yaitu atribut yang tidak dapat dibagi-bagi lagi menjadi atribut yang lebih sederhana (mendasar).
- Atribut komposit (*composite attribute*), yaitu atribut yang terdiri dari beberapa komponen dan dapat berdiri sendiri.
- Atribut berharga tunggal (*single-valued attribute*), yaitu atribut yang hanya mempunyai satu harga untuk suatu entitas tertentu.
- Atribut berharga ganda (*multi-valued attribute*), yaitu atribut yang dapat terdiri dari sekumpulan harga untuk suatu entitas tertentu.
- Null-value, yaitu atribut yang tidak mempunyai nilai dan tidak diketahui harganya.

2.1.5 DBMS (*Database Management System*)

2.1.5.1 Pengertian DBMS (*Database Management System*)

Menurut Silberschatz (1986, p1) DBMS terdiri dari kumpulan data yang saling berhubungan dan perangkat program untuk mengakses data. Tujuan utama dari DBMS adalah untuk menyediakan lingkungan yang tepat (*convenient*) dan efisien (*efficient*) untuk digunakan dalam mendapatkan kembali informasi dan menyimpannya ke dalam basis data.

Menurut Connolly & Begg (2002, pp16-20) DBMS adalah suatu sistem software yang memungkinkan user untuk mendefinisikan, menciptakan, mengatur, dan menyediakan akses ke basis data.

Dari definisi diatas, dapat disimpulkan bahwa DBMS adalah suatu sistem software yang memungkinkan user menciptakan, mengatur dan menyediakan lingkungan yang tepat dalam mendapatkan informasi dan menyimpannya ke dalam basis data.

Secara umum, DBMS menyediakan fasilitas:

1. DDL (*Data Definition Language*) adalah struktur basis data yang menggambarkan desain basis data secara keseluruhan. DDL merupakan bahasa yang memungkinkan pemakai untuk menggambarkan dan mengetahui nama entity, atribut, dan hubungan yang diminta untuk aplikasi, digunakan bersama dengan hubungan diantara batasan integrasi dan keamanan.
2. DML (*Data Manipulation Language*) adalah bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. DML membolehkan pemakai untuk memasukkan, mengubah, menghapus, dan mendapatkan kembali data yang ada di basis data. Seringkali di dalam operasi manipulasi data terdapat:
 - Penempatan data baru ke dalam basis data
 - Perubahan dari data yang disimpan dalam basis data
 - Mendapatkan kembali data yang dimuat dalam basis data
 - Penghapusan data dari basis data

Ada dua jenis dari DML, yaitu prosedural dan non-prosedural. Perbedaan dasar diantara keduanya adalah prosedural DML menentukan bagaimana keluaran dari pernyataan DML yang dihasilkan, sedangkan non-prosedural DML hanya menggambarkan keluaran apa yang dihasilkan. DML merupakan bahasa yang bertujuan untuk memudahkan pemakai dalam mengakses data sebagaimana digambarkan dalam basis data.

2.1.5.2 Komponen DBMS

Berikut ini merupakan komponen dari DBMS:

1. *Hardware*

Perangkat keras merupakan komponen dari *personal computer* (PC) dari network komputer.

2. *Software*

Komponen *software* terdiri dari DBMS itu sendiri dan program aplikasi, bersama dengan sistem operasi (*operating system*), termasuk jaringan *software* jika DBMS digunakan dalam jaringan.

3. *Data*

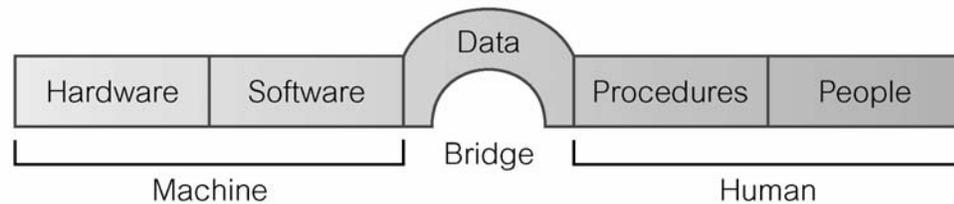
Digunakan oleh organisasi dan menjelaskan struktur dari data-data yang disebut *schema*.

4. *Procedures*

Prosedur mengacu kepada instruksi dan peraturan yang menentukan perancangan dan digunakan di dalam basis data.

5. *People*

Ada beberapa tipe pemakai terhadap suatu sistem basis data yaitu : *programmer, database administrators (DBA), system analysis.*



Gambar 2.5 Komponen DBMS

Sumber: Connolly & Begg(2002), p18

2.1.5.3 SQL (*Structured Query Language*)

SQL merupakan bahasa standar yang pertama dan sejauh ini merupakan bahasa basis data yang mendapatkan penerimaan yang besar. Idealnya, bahasa basis data akan memperbolehkan pemakai untuk:

- Membuat basis data dan struktur hubungan.
- Menampilkan tugas pengaturan data dasar, seperti memasukkan, modifikasi, dan menghapus data dari relasi.
- Menampilkan query yang sederhana maupun yang rumit.

SQL merupakan bahasa yang dirancang untuk menggunakan relasi untuk mengubah input menjadi output yang diperlukan. Sebagai sebuah bahasa, standar ISO SQL memiliki dua komponen utama :

- Data Definition Language (DDL) untuk menentukan struktur basis data dan mengatur akses ke data. Bentuk umum pernyataan DDL:

1. *Create Table*

Digunakan untuk membuat tabel basis data. Tabel terbentuk dari baris (tuple) dan kolom (atribut).

Sintaks : CREATE TABLE TableName

(ColumnName[,ColumnName]...[,PrimaryKeyDefinition][,Foreign
KeyDefinition])

2. *Alter Table*

Digunakan untuk melakukan perubahan dari struktur tabel, menambah atau menghapus suatu kolom.

Sintaks : ALTER TABLE TableName add|drop column data-type

3. *Drop Table*

Digunakan untuk menghapus tabel yang sudah ada.

Sintaks : DROP TABLE TableName

- Data Manipulation Language (DML) untuk memperbaharui data.

Bentuk umum pernyataan DML :

1. *Select*

Digunakan untuk mencari data yang ada di dalam basis data.

Sintaks : SELECT [DISTINCT | ALL]

{* | [columnExpression [AS newName]] [...]}

FROM TableName [alias] [...]

[WHERE condition]

[GROUP BY columnList] [HAVING condition]

[ORDER BY columnList]

2. *Insert*

Digunakan untuk menambah data baru ke dalam suatu tabel basis data.

Sintaks : INSERT INTO TableName ([columnList])
VALUES (data ValueList)

3. *Update*

Digunakan untuk memperbaharui data di dalam tabel basis data.

Sintaks : UPDATE TableName
SET columnName1 = dataValue1 [, columnName2 = dataValue2...]
[WHERE searchCondition]

4. *Delete*

Digunakan untuk menghapus satu atau beberapa baris dari sebuah tabel.

Sintaks : DELETE FROM TableName [WHERE searchCondition]

SQL merupakan bahasa yang sangat mudah untuk dipelajari:

- Merupakan bahasa *non-procedural*: kita mengkhususkan informasi apa yang kita perlukan, dibandingkan dengan bagaimana cara mendapatkannya. Dengan kata lain, SQL tidak memerlukan kita untuk mengkhususkan akses ke data.
- Seperti kebanyakan bahasa *modern*, SQL merupakan *free-format*, yang berarti bahwa bagian dari pernyataan tidak perlu diketik di lokasi tertentu pada layar.
- Struktur perintah terdiri dari standar kata Inggris seperti CREATE TABLE, INSERT, SELECT.

- SQL dapat digunakan oleh sejumlah pemakai meliputi database administrator, karyawan manajemen, pengembang aplikasi, dan banyak tipe *end-user* lainnya.

2.2 Teori-teori Khusus

Berikut ini merupakan teori-teori khusus tentang pembelian dan reparasi:

2.2.1 Pembelian

Pembelian merupakan kegiatan utama dalam suatu bisnis perusahaan selain penjualan. Berikut akan dibahas secara singkat tentang pembelian.

2.2.1.1 Pengertian pembelian

Menurut Clifton H.D (1994, p8) pembelian melibatkan prosedur untuk memastikan bahwa semua bahan, komponen, peralatan, perlengkapan dan benda lain yang dibutuhkan oleh perusahaan tersedia pada waktu yang tepat, tempat yang tepat dan harga yang tepat.

Menurut Mulyadi(1993, p301) sistem pembelian digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan

Menurut Assauri(1998, p159) pembelian merupakan salah satu fungsi yang penting dalam berhasilnya operasi suatu perusahaan.

Fungsi pembelian bergantung pada jenis benda yang dibeli, tapi biasanya mengikuti prosedur berikut:

- Dapatkan kutipan harga, tanggal pengiriman dan kualitas dari pemasok.
- Tempatkan pesanan dengan pemasok, awasi pengiriman.

- Periksa pengiriman barang untuk jenis, kualitas dan jumlah yang tepat.
- Kirim keterangan dari barang yang diterima ke bagian akuntansi untuk pembayaran ke pemasok.
- Periksa tagihan pemasok dan pernyataan untuk ketepatan sebelum membuat pembayaran.
- Kumpulkan informasi tentang harga pemasok, pengiriman dan kualitas.

Empat subsistem dari sistem pembelian:

1. Memilih pemasok. Pembeli memutuskan pemasok mana yang harus dipertimbangkan untuk pengisian kembali persediaan, dengan menggunakan data dari file pemasok dan mungkin dari sumber-sumber lain. file pemasok menyimpan data yang menjelaskan transaksi perusahaan dengan pemasok dimasa lampau dalam hal kualitas bahan, harga dan kemampuan untuk memenuhi tanggal pengiriman yang dijanjikan.
2. Mendapatkan komitmen lisan. Ketika pembeli menentukan pemasok mana yang merupakan calon terbaik, mereka dihubungi melalui telepon atau mungkin melalui jaringan dengan maksud mencapai suatu persetujuan.
3. Menyiapkan pesanan pembelian. Pembeli kemudian menggunakan terminalnya untuk mulai menyiapkan pesanan pembelian. Output yang lain adalah catatan pesanan pembelian yang ditambahkan pada file pesanan pembelian terbuka. File ini menyediakan suatu tingkat pengendalian atas pesanan pembelian yang hilang saat diposkan, atau

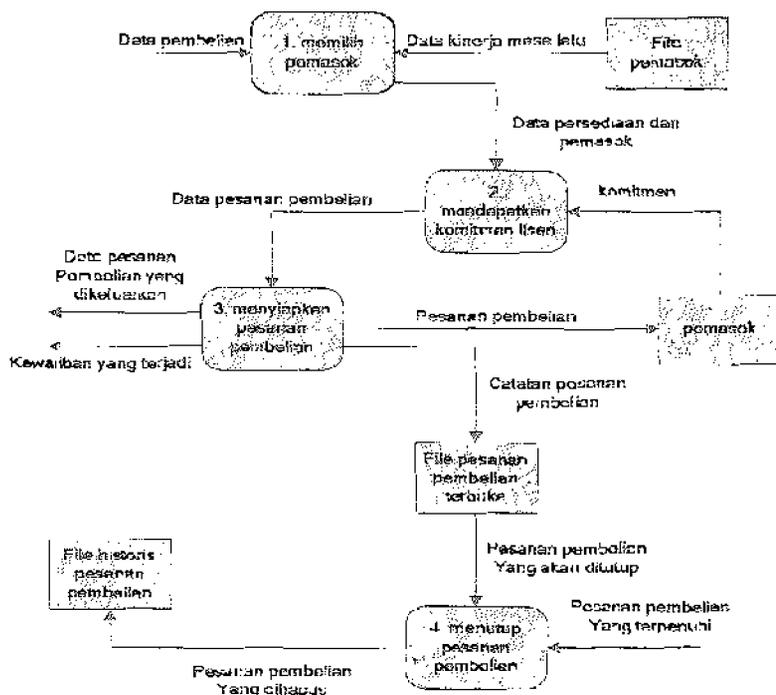
pengiriman yang tidak sesuai dengan yang dijanjikan. Saat file ini disimpan dalam medium magnetik, komputer dapat membaca tiap actatan secara harian dan memberitahukan kepad apembeli tiap pesana pembelian terbuka, yaitu yang seharusnya telah dipenuhi namun belum juga dipenuhi. Pembeli dapat membuat tindak lanjut dengan pemasok untuk menentukan alasannya.

Dua output lainnya adalah pemberitahuan ke sistem lain bahwa pembelian telah dilakukan. Arus data pesanan pembelian yang dikeluarkan, memberitahukan kepada sistem penerimaan bahwa pengiriman sedang berlangsung. Juga, arus data kewajiban yang terjadi memberitahukan sistem hutang dagang bahwa faktur dari pemasok akan diterima.

4. Menutup pesanan pembelian. Jika pesanan telah dipenuhi, perlu menghapus catatan dari file pesanan pembelian terbuka. Sinyal ini datang dari sistem penerimaan dalam bentuk arus data pesanan pembelian yang terpenuhi. Catatan pesanan pembelian yang dihapus disimpan dalam file historis pesanan pembelian.

Dua output lainnya adalah pemberitahuan ke sistem lain bahwa pembelian telah dilakukan. Arus data pesanan pembelian yang dikeluarkan, memberitahukan kepada sistem penerimaan bahwa pengiriman sedang berlangsung. Juga, arus data kewajiban yang terjadi memberitahukan sistem hutang dagang bahwa faktur dari pemasok akan diterima.

4. Menutup pesanan pembelian. Jika pesanan telah dipenuhi, perlu menghapus catatan dari file pesanan pembelian terbuka. Sinyal ini datang dari sistem penerimaan dalam bentuk arus data pesanan pembelian yang terpenuhi. Catatan pesanan pembelian yang dihapus disimpan dalam file historis pesanan pembelian.



Gambar 2.6 Sistem Pembelian

Sumber : McLeod, Raymond(1996), p19

- Catatan penerimaan barang. Merupakan suatu bukti catatan jumlah barang. Catatan ini dibuat pada waktu barang diterima dan kemudian divalidasikan setelah barang diperiksa.
- Tagihan pembelian. Tagihan ini diterima dari pemasok setelah barang telah diterima. Tagihan ini harus mengandung keterangan yang cukup yang berhubungan dengan barang yang diterima dan ada di pesanan pembelian.
- Rekomendasi Pembayaran. Merupakan dokumen yang dikirim ke pemasok merekomendasikan pembayaran yang harus dibayarkan. Informasinya adalah jumlah uang, tanggal dan metode pembayaran.
- Cek. Cek yang disertakan bersama dengan rekomendasi pembayaran harus ditarik kecuali pembayaran dilakukan melalui transfer bank.
- Transfer kredit. Daftar dikirim ke bank perusahaan memerintahkan mereka untuk mengirim uang ke account pemasok.
- Analisis pembelian. Dalam beberapa kasus, sangatlah penting untuk menganalisis pembelian perusahaan dalam beberapa cara. Tiga alasan untuk hal ini adalah sebagai berikut:
 1. Untuk mengukur keefisienan pemasok.
 2. Untuk membiayai pembelian departemen.
 3. Untuk membiayai pembelian dalam menyelesaikan pekerjaan, yaitu biaya pekerjaan.
- Record pembelian. Record harus disimpan untuk setiap pemasok untuk menunjukkan keterangan dari accountnya, yaitu buku besar pembelian.

Informasi utama dalam tiap record adalah nomor account pemasok, nama, keseimbangan dan transaksi (tagihan dan rekomendasi).

2.2.1.3 Retur Pembelian

Barang yang sudah diterima dari pemasok adakalanya tidak sesuai dengan barang yang dipesan menurut surat order pembelian. Ketidaksesuaian tersebut terjadi kemungkinan karena barang yang diterima tidak cocok dengan spesifikasi yang tercantum dalam surat order pembelian. Barang mengalami kerusakan dalam pengiriman atau barang diterima melewati tanggal pengiriman yang dijanjikan oleh pemasok. Sistem retur pembelian digunakan dalam perusahaan untuk pengembalian barang yang sah diteliti kepada pemasoknya.

2.2.2 Pengertian Reparasi

Menurut Poerwadarminta (1993, p820) reparasi adalah pembetulan apa-apa yang rusak. Saat ini, mobil telah menjadi sebuah alat transportasi yang sangat dibutuhkan oleh masyarakat. Adanya kendaraan ini, memudahkan kita untuk bepergian dari satu tempat ke tempat lain hanya dalam waktu yang relatif singkat. Sayangnya, banyak sekali orang yang tidak mengetahui cara merawat dan memperbaiki kerusakan mobil. Oleh karena itu, jika mobil rusak, orang lebih suka membawanya ke bengkel.

Menurut Poerwadarminta (1993, p119) bengkel adalah tempat memperbaiki mobil, sepeda, dll.