

Basis Data

Oleh : Diana Effendi, ST.,MT

(Digunakan di lingkungan sendiri, sebagai buku ajar
mata kuliah Basis Data)



Fakultas Teknik dan Ilmu Komputer
Program Studi Sistem Informasi
Universitas Komputer Indonesia

1. Pertemuan 1

1.1 Perkenalan Matakuliah

a. Deskripsi Matakuliah

Mata kuliah ini memberikan penjelasan tentang dasar-dasar perancangan serta implementasi basis data, dimana mata kuliah ini merupakan teori basis data yang sangat mendasar tanpa memerlukan prasyarat pengetahuan tentang basis data sebelumnya. Materi mata kuliah ini difokuskan pada tiga hal pokok, yaitu: dasar-dasar basis data, model data dan bagaimana membuat model data yang baik. Selain itu, mata kuliah ini juga membahas contoh-contoh aturan yang berlaku di lapangan yang terefleksikan dalam perancangan basis data. Sejumlah aspek lain yang relevan, seperti DBMS, SQL, aplikasi basis data dan perkembangan teknologi terbaru juga dibahas dalam mata kuliah ini.

b. Sistem Perkuliahan

- a. Perkuliahan diselenggarakan 14 kali pertemuan (2 SKS)
- b. Wajib kehadiran Mahasiswa 80% (-3 kali tidak masuk)
- c. Materi perkuliahan akan diberikan salinannya kepada Mahasiswa
- d. Mahasiswa dianjurkan membawa flashdisk
- e. Batas keterlambatan 15 menit setelah perkuliahan dimulai
- f. Mahasiswa diperbolehkan berkonsultasi dengan dosen; mengenai materi perkuliahan secara personal atau kelompok di luar jam perkuliahan (tatap muka; via email; kuliah online)
- g. Mengikuti tata tertib Lab
- h. Tidak diperbolehkan menggunakan perangkat komunikasi selama perkuliahan (setting silent/vibrate)
- i. Bersikap sopan dan tidak mengganggu keberlangsungan perkuliahan
- j. Tersedia waktu Shalat bagi yang beragama Islam.

1.2. Pengantar Basis Data

Dari awal penggunaan komputer, penyimpanan dan manipulasi data merupakan focus utama aplikasi. Pada awal tahun 1960, Charles Bachman di perusahaan General Electric mendesain generasi pertama DBMS yang disebut Penyimpanan Data Terintegrasi (*Integrated Data Store*). Dasar untuk model data jaringan dibentuk lalu distandardisasi oleh *Conference on Data System Language*(CODASYL). Kemudian, Bachman menerima ACM Turing Award (Penghargaan semacam nobel pada ilmu komputer) di tahun 1973.

1.3 Sejarah Kemunculan Basis Data

Pada akhir tahun 1960-an, IBM mengembangkan system manajemen informasi (*Information Manajemen System*) DBMS. IMS dibentuk dari representasi data pada kerangka kerja yang disebut model data hierarki. Dalam waktu yang sama, hasil kerja sama antara IBM dengan perusahaan penerbangan Amerika mengembangkan system SABRE. System SABRE memungkinkan user mengakses data yang sama pada jaringan computer.

- Tahun 1970

Pada tahun 1970, Edgar Codd di laboratorium penelitian di San Jose mengusulkan suatu representasi data baru yang disebut model data relational. Pada tahun 1980, model relasional menjadi paradigm DBMS paling dominan. Bahasa *query* SQL dikembangkan untuk basisdata relasional sebagai bagian proyek Sistem R dari IBM. SQL di distandardisasi di akhir tahun 1980 dan SQL-92 diadopsi oleh American National Standards

Institute (ANSI) dan International Standards Organization (ISO). Program yang digunakan untuk eksekusi bersamaan dalam basisdata disebut transaksi. User menulis programnya, dan bertanggung jawab menjalankan program secara bersamaan terhadap DBMS. Pada tahun 1999, James Gray memenangkan Turing award untuk kontribusinya pada manajemen transaksi dalam DBMS.

- Tahun 1980

Pada akhir tahun 1980 dan permulaan tahun 1990, banyak bidang system basisdata dikembangkan. Penelitian dibidang basisdata meliputi bahasa *query* yang *powerful*, model data yang lengkap, dan penekanan pada dukungan analisis data yang kompleks semua bagian organisasi. Beberapa vendor (misalnya IBM, DB2, Oracle8, dan Informix UDS) memperluas sistemnya dengan kemampuan menyimpan tipe data baru misalnya image dan text serta kemampuan query yang kompleks. System khusus dikembangkan banyak vendor untuk membuat *data warehouse* dan mengonsolidasi data beberapa basisdata.

Suatu fenomena menarik adalah munculnya *enterprise resource planning* (ERP) dan *management resource planning* (MRP), yang menambah lapisan substansial dari fitur berorientasi aplikasi pada DBMS utama. Paket yang digunakan secara luas meliputi Baan, Oracle, PeopleSoft, SAP, dan Siebel. Paket tersebut mengidentifikasi kumpulan tugas umum (misalnya manajemen inventori, perencanaan sumber daya manusia, dan analisis keuangan) yang dihadapi oleh sejumlah besar organisasi dan menyediakan lapisan aplikasi umum untuk melaksanakan tugas. Data disimpan dalam DBMS relasional. Kemudian, lapisan aplikasi dapat disesuaikan pada perusahaan berbeda sehingga biaya keseluruhan perusahaan menjadi lebih rendah dibanding biaya pembuatan lapisan aplikasi dari awal. Lebih jauh, DBMS memasuki dunia internet. Saat generasi pertama, web site menyimpan datanya secara eksklusif dalam file system operasi. Pada saat ini, DBMS dapat digunakan untuk menyimpan data yang dapat diakses melalui web browser. Query dapat dibuat melalui form web dan format jawabannya

dengan menggunakan *markup language semisal HTML* untuk mempermudah tampilan pada browser. Semua vendor basisdata menambah fitur ini untuk DBMS mereka.

Manajemen basisdata mempertimbangkan pentingnya suatu data bersifat online dan dapat diakses melalui jaringan computer. Saat ini, bidang seperti ini diwujudkan dalam basisdata multimedia, video interaktif, perpustakaan digital, proyek ilmuwan seperti proyek pemetaan, proyek system obeservasi bumi milik NASA, dan lain sebagainya (Ramakrishnan and Gehrke, 2003).

1.4 Definisi

Pengertian basis data dan sistem basis data – Basis data adalah, basis data terdiri dari 2 (dua) kata, yaitu kata Basis dan Data. Basis bisa di artikan sebagai markas ataupun gudang, tempat berkumpul. Sedangkan data yaitu kumpulan fakta dunia nyata yang mewakili suatu objek, seperti manusia, barang, dan lain-lain yang direkam ke dalam bentuk angka, bentuk huruf, simbol, teks, bunyi, gambar atau juga kombinasinya.

1.5 Aplikasi Sistem Basis Data

1. Microsoft Access

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/programmer yang mahir

dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

2. Ms SQL Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (Tabular Data Stream). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (Open Database Connectivity), dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data mirroring dan clustering. Pada versi sebelumnya, MS SQL Server 2000 terserang oleh cacing komputer SQL Slammer yang mengakibatkan kelambatan akses Internet pada tanggal 25 Januari 2003.

3. Oracle

Basis data Oracle adalah basis data relasional yang terdiri dari kumpulan data dalam suatu sistem manajemen basis data RDBMS. Perusahaan perangkat lunak Oracle memasarkan jenis basis data ini untuk bermacam-macam aplikasi yang bisa berjalan pada banyak jenis dan merk perangkat keras komputer (platform).

Basis data Oracle ini pertama kali dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates lewat perusahaan konsultasinya bernama Software Development Laboratories (SDL) pada tahun 1977. Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation sampai sekarang.

4. MySql

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

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

5. Postgre SQL

Postgre SQL merupakan tool untuk membuat database server yang bersifat open source yang memiliki lisensi GPL (General Public License). Postgre SQL mendukung bahasa pemrograman seperti : SQL, C, C++, Java, PHP dan lainnya. Dengan lisensi GPL, PostgreSQL dapat digunakan, dimodifikasi dan didistribusikan oleh setiap orang tanpa perlu membayar lisensi (free of charge) baik untuk keperluan pribadi, pendidikan maupun komersil. Kebanyakan PostgreSQL tersedia untuk pengguna Linux.

6. Firebird

Firebird (juga disebut FirebirdSQL) adalah sistem manajemen basisdata relasional yang menawarkan fitur-fitur yang terdapat dalam standar ANSI SQL-99 dan SQL-2003. RDBMS ini berjalan baik di Linux, Windows, maupun pada sejumlah platform Unix. Firebird di diarahkan dan di-maintain oleh FirebirdSQL Foundation. Ia merupakan turunan dari Interbase versi open source milik Borland. Modul-modul kode baru ditambahkan pada Firebird dan berlisensi di bawah Initial Developer's Public License (IDPL), sementara modul-modul aslinya dirilis oleh Inprise berlisensi di bawah InterBase Public License 1.0. Kedua lisensi tersebut merupakan versi modifikasi dari Mozilla Public License 1.1.

1.6 Objektif Basis Data

Telah disebutkan di awal bahwa tujuan awal dan utama dalam pengelolaan data dalam sebuah basis data adalah agar kita dapat memperoleh/menemukan kembali data (yang kita cari) dengan mudah dan cepat. Disamping itu, pemanfaatan basis data untuk pengelolaan data, juga memiliki tujuan-tujuan lain.

Secara lebih lengkap, pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti buku ini :

Kecepatan dan Kemudahan (Speed);Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika kita menyimpan data secara manual (non elektronik) atau secara elektronik (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk spread sheet atau dokumen teks biasa)

Efisiensi Ruang Penyimpanan (Space);Karena keterkaitan yang erat antar kelompok data dalam sebuah basis data, maka redundansi (pengulangan) data pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruang penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah

redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

Keakuratan (Accuracy); Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (constraint) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

Ketersediaan (Availability); Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi kita gunakan, dapat kita atur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi off-line) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan off-line (seperti removable disk atau tape). Di sisi lain, karena kepentingan pemakaian data, sebuah basis data dapat memiliki data yang tersebar di banyak lokasi geografis. Data nasabah sebuah bank, misalnya, dipisah-pisah dan disimpan di lokasi yang sesuai dengan keberadaan nasabah. Dengan pemanfaatan teknologi jaringan komputer, data yang berada di suatu lokasi/cabang, dapat juga diakses (menjadi tersedia/available) bagi lokasi/cabang lain.

Kelengkapan (Completeness); Lengkap/tidaknya data yang kita kelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai yang lain belum tentu berpendapat sama. Atau, yang sekarang dianggap sudah lengkap, belum tentu di masa yang akan datang juga demikian. Dalam sebuah basis data, di samping data kita juga harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi dari tiap objek, seperti struktur file/tabel atau indeks). Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah record-record data, tetapi juga dapat

melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

Keamanan (Security); memang ada sejumlah (aplikasi) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan dengan ketat. Dengan begitu kita dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

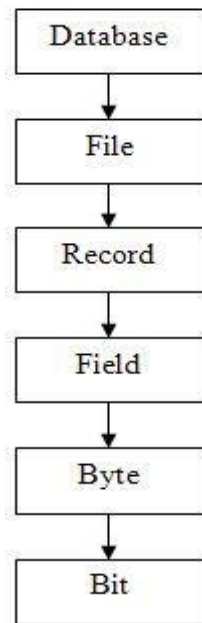
Kebersamaan Pemakaian (Sharability); Pemakai basis data seringkali tidak terbatas pada satu pemakai saja, atau di satu lokasi saja atau oleh satu sistem/aplikasi saja. Data pegawai dalam basis data kepegawaian, misalnya, dapat digunakan oleh banyak pemakai, dari sejumlah departemen dalam perusahaan atau oleh banyak sistem (sistem penggajian, sistem akuntansi, sistem inventori, dan sebagainya). Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari (karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi deadlock (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

1.7 Hirarki Data

Database merupakan kumpulan dari data yang saling berinteraksi satu dengan yang lainnya, tersimpan dalam perangkat keras komputer dan digunakan diperangkat lunak untuk memanipulasi. Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya.

Menurut Jogiyanto Hartono, MBA, Ph. D, "Pengenalan Komputer", (1999:71), database adalah suatu sistem informasi yang menginterasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Tujuan database adalah untuk menentukan data yang dibutuhkan dalam sistem, sehingga

informasinya yang dihasilkan dapat dipenuhi dengan baik. Adapun bentuk dari hirarki sebuah database dapat dilihat dari gambar 3.10 adalah sebagai berikut.



Gambar 3.13 *Hirarki* data dalam database
sumber : Analisa dan Desain, Jogiyanto,2000

Hirarki data dalam database mulai dari yang terbesar ke yang terkecil yaitu :

1. Database

Suatu database menggambarkan data yang saling berhubungan antara satu dengan yang lainnya.

2. File

Suatu file menggambarkan suatu kesatuan data yang sejenis, dimana kumpulan dari file membentuk suatu database.

3. Record

Suatu record menggambarkan suatu unit data individu yang tertentu dimana kumpulan dari record membentuk suatu file.

4. Field

Suatu field menggambarkan suatu attribute dari record, dimana kumpulan field membentuk suatu record.

5. Byte

Attribute dari field berupa huruf yang membentuk nilai dari sebuah field.

6. Bit

Merupakan bagian terkecil dari data secara keseluruhan yaitu berupa karakter ASCII (American Standar Code Form Information Interchange). 0 (nol) adalah satu yang merupakan komponen pembentuk byte.

2. Pertemuan 2

2.1 Komponen Sistem Basis Data

Komponen Sistem Basis Data terdiri dari 6 Komponen , yakni :

1. Hardware

Biasanya berupa perangkat komputer standar, media penyimpanan sekunder dan media komunikasi untuk sistem jaringan.

2. Operating System

Yakni merupakan perangkat lunak yang memfungsikan, mengendalikan seluruh sumber daya dan melakukan operasi dasar dalam sistem komputer. Harus sesuai dengan DBMS yang digunakan.

3. Database

Yakni basis data yang mewakili sistem tertentu untuk dikelola. Sebuah sistem basis data bisa terdiri dari lebih dari satu basis data.

4. DBMS (Database Management System)

Perangkat lunak yang digunakan untuk mengelola basis data. Contoh kelas sederhana: dBase, Foxbase, Rbase, MS. Access, MS. Foxpro, Borland Paradox. Contoh kelas kompleks: Borland-Interbase, MS. SQL Server, Oracle, Informix, Sybase.

5. User (Pengguna Sistem Basis Data)

Orang-orang yang berinteraksi dengan sistem basis data, mulai dari yang merancang sampai yang menggunakan di tingkat akhir.

6. Optional Software

Perangkat lunak pelengkap yang mendukung. Bersifat opsional.

2.2 Mengenal DBMS

a. Pengertian DBMS

DBMS adalah singkatan dari “Database Management System” yaitu sistem penorganisasian dan sistem pengolahan Database pada komputer. DBMS atau database management system ini merupakan perangkat lunak (software) yang dipakai untuk membangun basis data yang berbasis komputerisasi. DBMS (Database Management system) ini juga dapat membantu dalam memelihara serta pengolahan data dalam jumlah yang besar, dengan menggunakan DBMS bertujuan agar tidak dapat menimbulkan kekacauan dan dapat dipakai oleh user sesuai dengan kebutuhan.

DBMS ialah perantara untuk user dengan basis data, untuk dapat berinteraksi dengan DBMS dapat memakai bahasa basis data yang sudah ditentukan oleh perusahaan DBMS. Bahasa basis data umumnya terdiri dari berbagai macam intruksi yang diformulasikan sehingga intruksi tersebut dapat diproses oleh DBMS. Perintah atau intruksi tersebut umumnya ditentukan oleh user.

Adapun bahasa yang digunakan dibagi kedalam 2 (dua) macam diantaranya sebagaimana di bawah ini:

1. DDL (Data Definition Language)

Yang pertama adalah bahasa DDL atau kepanjangannya Data Definition Language, yaitu dipakai untuk menggambarkan desain dari basis data secara menyeluruh. DDL (Data Definition Language) dapat dipakai untuk membuat tabel baru, memuat indeks, maupun mengubah tabel. Hasil dari kompilasi DDL akan disimpan di kamus data. Itulah definisi dari DDL.

2. DML (Data Manipulation Language)

Dan yang kedua adalah DML atau kepanjangannya Data Manipulation Language, yaitu dipakai untuk memanipulasi

dan pengambilan data pada suatu basis data, misalnya seperti penambahan data yang baru ke dalam suatu basis data, menghapus data pada suatu basis data dan mengubah data pada suatu basis data. Itulah definisi dari DML.

b. Macam-macam atau contoh DBMS (Database management system)

Adapun beberapa contoh dari DBMS, diantaranya seperti di bawah ini:

1. MySQL

Kelebihannya:

Free/gratis.

Selalu stabil dan cukup tangguh.

Keamanan yang cukup baik.

Sangat mendukung transaksi, dan dukungan dari banyak komunitas.

Sangat fleksibel dengan berbagai macam program.

Perkembangan yang cepat.

Kekurangannya:

Kurang mendukung koneksi bahasa pemrograman misalnya seperti Visual Basic (VB), Foxpro, Delphi sebab koneksi ini dapat menyebabkan field yang dibaca harus sesuai dengan koneksi bahasa pemrograman visual tersebut.

Data yang dapat ditangani belum besar dan belum mendukung windowing Function.

2. Oracle

Kelebihannya:

Terdapat beragam fitur yang bisa memenuhi tuntutan fleksibilitas dari organisasi atau perusahaan yang besar. Bisa mendayagunakan lebih dari satu server dan penyimpanan data dengan cukup mudah. Performa pemrosesan transaksi yang sangat tinggi. Kekurangannya:

Pemakaiannya membutuhkan dana atau biaya karena mahal dan diperlukan DBA yang cukup handal sebab DBMS ini cukup rumit.

3. Microsoft SQL server

Kelebihannya:

DBMS ini sangat cocok untuk perusahaan mikro, menengah hingga perusahaan besar karena mampu mengelola data yang besar.

Mempunyai kelebihan untuk men-manage user serta tiap user-nya dapat diatur hak aksesnya terhadap pengaksesan database oleh DBA.

Tingkat pengamanan datanya sangat baik.

Dapat melakukan atau memiliki back-up, recovery, dan rollback data.

Kelebihan lainnya mempunyai kemampuan membuat database mirroring dan juga culustering.

Kekurangannya:

Hanya bisa berjalan pada platform OS (Operasi system) Microsoft windows.

Perangkat lunak (software) ini berlisensi dan tentunya pemakaiannya membutuhkan biaya yang tergolong cukup mahal.

Itulah beberapa contoh dari DBMS.

Fungsi DBMS :

Data definition, DBMS harus dapat mengolah pendefinisian data

Data manipulation, DBMS harus dapat menangani permintaan dari user untuk mengakses data

Data security and integrity, DBMS harus dapat memeriksa keamanan dan integriti data yang didefinisikan oleh Database Administrator.

Data recovery and concurrency, DBMS harus dapat menangani kegagalan pengaksesan database.

Data dictionary, DBMS harus dapat menyimpan informasi yang menggambarkan data dalam database, biasa disebut metadata (data of data).

Performance, DBMS harus dapat menangani semua fungsi seefisien mungkin.

Keuntungan DBMS :

tidak terjadi kerangkapan data

data lebih konsisten

data dapat digunakan bersama-sama

keamanan data dapat terjamin

integritas data terpelihara

data independen

Dan inilah komponen DBMS (Database Management System)

DBMS biasanya mempunyai komponen fungsional (modul), diantaranya sebagaimana di bawah ini:

- File Manager adalah mengelola ruang didalam suatu disk dan juga struktur data yang digunakan untuk merepresentasikan informasi yang tersimpan didalam suatu disk.
- Database Manager adalah menyediakan interface antar data low – level yang terdapat pada basis data dengan program aplikasi serta query yang diberikan ke suatu sistem.

- Query Processor adalah menterjemahkan perintah dalam bahasa query ke intruksi low – level yang dapat dimengerti database manager.
- DML Precompiler adalah mengkonversi pernyataan atau perintah DML, yang ditambahkan dalam suatu program aplikasi kepemangin prosedur normal dalam bahasa induk.
- DDL Compiler adalah yang mengkonversi berbagai perintah DDL ke dalam sekumpulan tabel yang mengandung metadata.

2.3 Abstraksi Data

Abstraksi data merupakan tingkatan/level dalam bagaimana melihat data dalam sebuah sistem basis data.

Ada 3 level abstraksi data :

- Level Fisik (Physical Level) –Internal Level
- Level Logik/Konseptual (Conceptual Level)
- Level Penampakan (View Level) –External Level

Physical Level –Internal Level merupakan level terendah, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada level ini, pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri.

Conceptual Level Menggambarkan data apa yang sebenarnya disimpandalam basis data dan hubungannya dengan data yanglain.

View Level –External Level Merupakan level tertinggi, hanya menunjukkansebagiaan dari basis data sesuai dengan kebutuhanuser, bagi user yang menggunakan terasa sebagai satukesatuan data yang kompak.

2.4 Bahasa Basis Data

Bahasa Basis Data dipilah ke dalam 2 bentuk yaitu ,

1. Data Definition Language (DDL)

Dengan bahasa ini kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel dsb.

2. Data Manipulation Language (DML).

Berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data.

Berupa:

- penyisipan/penambahan data baru (insert)
- penghapusan data (delete)
- pengubahan data (update)

3 Pertemuan 3

3.1 Model E-R

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

Menurut salah satu para ahli, Brady dan Loonam (2010), Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analysts dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

3.2 Model Relasional

a. Pengertian Model Data Relasional

Model Relasional merupakan model data yang paling banyak digunakan saat ini. Hal ini disebabkan oleh bentuknya yang sederhana dibandingkan dengan model jaringan/network atau model hirarki. Bentuk yang sederhana ini membuat pekerjaan seorang programmer

menjadi lebih mudah, yaitu dalam melakukan berbagai operasi data (query, insert, update, delete, dan lainnya).

Model Data Relasional adalah model basis data yang menggunakan tabel dua dimensi, yang terdiri dari baris dan kolom untuk menggambarkan sebuah berkas data.

b. Keuntungan Model Data Relasional

Bentuknya sederhana sehingga mudah dalam penggunaannya.

Mudah melakukan berbagai operasi data (query, update/edit, delete).

Istilah-istilah dalam Model Data Relasional :

- Relasi yaitu sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris.
- Atribut yaitu kolom pada sebuah relasi.
- Tupel yaitu baris pada sebuah relasi.

- Domain yaitu kumpulan nilai yang valid untuk satu atau lebih atribut
- Derajat yaitu jumlah atribut dalam sebuah relasi (jumlah field)
- Cardinality yaitu jumlah tupel dalam sebuah relasi (jumlah record)
- Relational Key
- Super key

Satu/kumpulan atribut yang secara unik mengidentifikasi sebuah tupel di dalam relasi (satu atau lebih field yang dapat

dipilih untuk membedakan antara 1 record dengan record lainnya).

- Candidate key

Atribut di dalam relasi yang biasanya mempunyai nilai unik (super key dengan jumlah field yang paling sedikit)

- Primary key

Candidate key yang dipilih untuk mengidentifikasi tupel secara unik dalam relasi

- Alternate key

Candidate key yang tidak dipilih sebagai primary key

- Foreign key

Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah relasi tetapi pada relasi lain atribut tersebut hanya sebagai atribut biasa

- Relational Integrity Rules

- Null

Nilai suatu atribut yang tidak diketahui dan tidak cocok untuk baris (tuple) tersebut

- Entity Integrity

Tidak ada satu komponen primary key yang bernilai null.

- Referential Integrity

Suatu domain dapat dipakai sebagai kunci primer bila merupakan atribut tunggal pada domain yang bersangkutan.

4 Pertemuan 4

4.1 Diagram E-R

1. Entitas

Entitas adalah objek dalam dunia nyata yang dapat dibedakan dengan objek lain, sebagai contoh mahasiswa, dosen, departemen. Entitas terdiri atas beberapa atribut sebagai contoh atribut dari entitas mahasiswa adalah nim, nama, alamat, email, dll. Atribut nim merupakan unik untuk mengidentifikasi / membedakan mahasiswa yg satu dengan yg lainnya. Pada setiap entitas harus memiliki 1 atribut unik atau yang disebut dengan primary key.

2. Relasi

Relasi adalah hubungan antara beberapa entitas. sebagai contoh relasi antar mahasiswa dengan mata kuliah dimana setiap mahasiswa bisa mengambil beberapa mata kuliah dan setiap mata kuliah bisa diambil oleh lebih dari 1 mahasiswa. relasi tersebut memiliki hubungan banyak ke banyak. Berikut adalah contoh ERD.

3. Atribut

Atribut adalah Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

- Ada dua jenis Atribut :

- 1) Identifier (key) digunakan untuk menentukan suatu entity secara unik (primary key).

- 2) Descriptor (nonkey attribute) digunakan untuk menspesifikasikan karakteristik dari suatu entity yang tidak unik.

4. Key

5. Kardinalitas

Kardinalitas menyatakan jumlah himpunan relasi antar entitas. pemetaan kardinialit terdiri dari :

- one-to-one : sebuah entitas pada A berhubungan dengan entitas B paling banyak 1 contoh diatas relasi pegawai dan departemen dimana setiap pegawai hanya bekerja pada 1 departemen
- one-to-many : sebuah entitas pada A berhubungan dengan entitas B lebih dari satu contoh diatas adalah 1 departemen memiliki banyak pegawai
- many-to-many : sebuah entitas pada A berhubungan dengan entitas B lebih dari satu dan B berhubungan dengan A lebih dari satu jagan contoh diatas adalah relasi mahasiswa dengan mata kuliah.

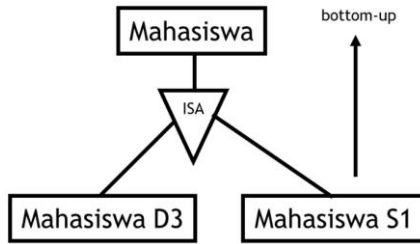
6. Varian Entitas/ Relasi

Varian Entitas yaitu :

- Strong Entity: Entity yang dapat berdiri sendiri tidak bergantung pada Entity lain.
Contoh : Mahasiswa
- Weak Entity: Entity yang bergantung pada Strong Entity. Berisi entitas yang kemunculannya tergantung pada eksistensinya dalam sebuah relasi.
Contoh : entitas Mahasiswa -orang tua & hobby, pegawai – tanggungan.

7. Generalisasi

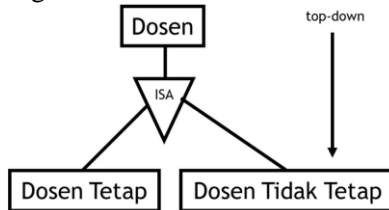
Generalisasi adalah proses perancangan yang bersifat bottom-up. Pembuatan entitas yang bersifat umum dimulai dari tipe-tipe entitas yang bersifat khusus.



8. Spesialisasi

Proses perancangan yang bersifat top-down.

Pembuatan satu atau beberapa entitas yang bersifat khusus (subtipe) dilakukan dari entitas yang bersifat umum terlebih dahulu (supertipe).



9. Agregasi

Sebuah relasi terbentuk tidak hanya dari entitas tapi terkadang juga mengandung unsur dari relasi yang lain.

Agregasi menggambarkan sebuah himpunan relasi yang secara langsung menghubungkan sebuah himpunan entitas dengan sebuah himpunan relationship dalam diagram ER.

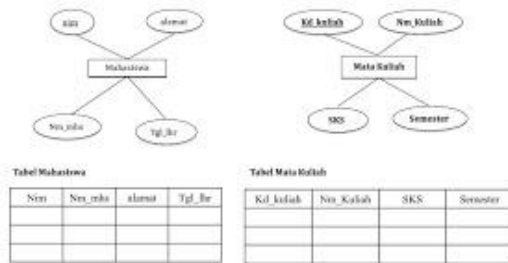
5 Pertemuan 5

5.1 Transformasi Model Data ke Basis Data Fisik

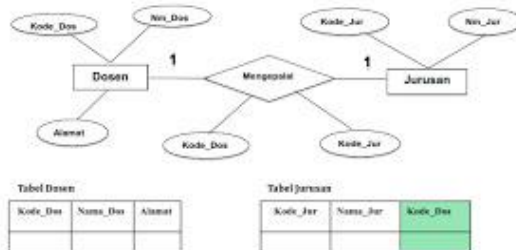
Transformasi merupakan perubahan dari suatu bentuk ke bentuk yang lain. Dalam pembahasan kita berubah bentuk menjadi basis data fisik yang kita gunakan untuk membangun suatu sistem basis data, yakni ERD (Entity Relationship Diagram) ditransformasikan dalam bentuk basis data secara fisik.

Komponen ERD (Entity Relationship Diagram) ditransformasikan dalam bentuk tabel yang merupakan komponen utama pembentukan basis data. Atribut yang terdapat pada masing-masing entitas akan dinyatakan sebagai field atau kolom dari tabel yang sesuai. Teknis untuk melakukan transformasi ERD ke dalam basis data fisik.

Setiap entitas akan diimplementasikan dalam bentuk sebagai sebuah tabel. contohnya seperti berikut ini :



5.2 Relasi dengan derajat Relasi 1-1 yang menghubungkan 2 buah himpunan entitas yang direpresentasikan dalam bentuk penambahan/penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas. contohnya seperti berikut ini :

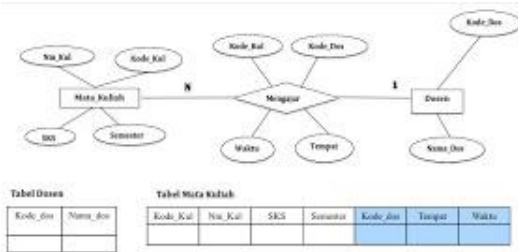


Gambar 4. Transformasi entitas dan relasinya yang mempunyai drajat relasi 1-1

Relasi dengan derajat relasi 1-N yang menghubungkan 2 buah himpunan entitas, juga akan direpresentasikan dalam bentuk

pemberian/pencantuman atribut key dari himpunan entitas berderajat 1 ke tabel yang mewakili himpunan entitas berderajat N.

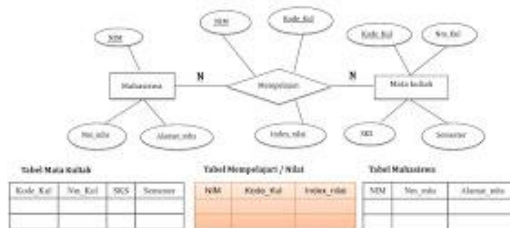
5.3 Atribut key dari himpunan entitas berderajat 1 menjadi atribut tambahan bagi himpunan entitas berderajat N.
 Contoh :



Gambar 5. Transformasi entitas dan relasinya yang mempunyai drajat relasi N-1

5.4 Relasi dengan derajat relasi N-N yang menghubungkan 2 buah himpunan entitas, diwujudkan dalam bentuk tabel khusus yang memiliki field (tepatnya foreign key) yang berasal dari key-key dari himpunan Entitas yang dihubungkannya.

Contoh :

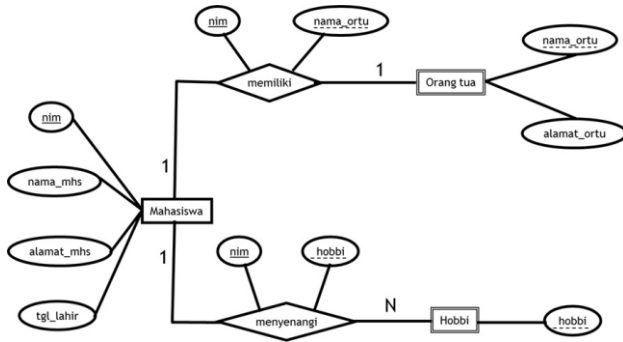


Gambar 6. Transformasi entitas dan relasinya yang mempunyai drajat relasi N-N

5.5 Entitas lemah/sub entitas

Himpunan entitas lemah tidak dapat berdiri sendiri tanpa entitas kuat. Dalam mentransformasikan dari ERD ke dalam bentuk basis data fisik keduanya sama akan terbentuk sebuah tabel. Namun yang membedakan adalah jika entitas kuat jika ditransformasikan ke dalam basis data fisik langsung menjadi tabel memiliki atribut key yang bertindak sebagai primary key dan atribut yang lain secara utuh, maka entitas lemah hanya dapat di transformasikan dalam bentuk tabel dengan menyertakan primary key dari entitas kuat dan atribut yang dimiliki oleh entitas lemah itu sendiri.

Contoh :



6 Pertemuan 6

6.1 Implementasi Basis Data

a. Pengertian

Implementasi basis data merupakan suatu tahapan dalam proses perancangan basis data. Tahap ini merupakan implementasi dari hasil pemodelan logical dan fisik. Bahasa perintah yang digunakan, baik itu untuk definisi data ataupun penyimpanan data harus sesuai dengan DBMS yang dipilih. Implementasi penyusunan basis data dimulai dari pembuatan berkas-berkas data (tabel-tabel entity) kosong yang akan digunakan untuk menyimpan data dalam basis data. Kemudian dilanjutkan dengan pemasukan data untuk tiap instan entity.

b. Tahapan Implementasi Basis Data

Tahap implementasi basis data merupakan upaya untuk membangun basis data fisik yang ditempatkan dalam memori sekunder (disk) dengan bantuan DBMS (Database Management System) yang kita pilih. Tahap implementasi basis data diawali dengan melakukan transformasi dari model data yang telah selesai dibuat ke skema/struktur basis data sesuai dengan DBMS yang dipilih. Secara umum, sebuah diagram E-R akan direpresentasikan menjadi sebuah basis data secara fisik. Sedang komponen-komponen Diagram E-R yang berupa himpunan entitas dan himpunan relasi akan ditransformasikan menjadi tabel-tabel (file-file data) yang merupakan komponen utama pembentuk basis data. Selanjutnya atribut-atribut yang melekat pada masing-masing himpunan entitas dan himpunan relasi akan dinyatakan sebagai field-field dari tabel-tabel yang sesuai.

Performansi basis data ditentukan oleh :

- Kualitas dan bentuk perancangan basis data
- Kualitas mesin / komputer
- Platform yang dipilih
- Sistem operasi
- DBMS yang digunakan

c. Pengkodean/Abstraksi Data

Data yang dilihat oleh pemakai awam (end-user) bisa berbeda dengan bagaimana data / informasi itu disimpan. Apa yang dilihat oleh end-user bisa jadi merupakan hasil pengolahan yang tidak disimpan sama sekali dalam basis data, atau bisa dinyatakan dalam bentuk lain. Alasan untuk membuat suatu pengkodean adalah untuk efisiensi ruang penyimpanan. Dari pemakaiannya, ada dua bentuk pengkodean :

1. Eksternal (User-Defined Coding)

Mewakili pengkodean yang telah digunakan secara terbuka dan dikenal dengan baik oleh pemakai awam. Contoh : Nomor mahasiswa dan Kode matakuliah → sudah dikenal baik oleh pemakai awam.

2. Internal (System Coding)

Menggambarkan bagaimana data disimpan dalam kondisi sebenarnya, sehingga lebih berorientasi pada mesin. Ada tiga bentuk pengkodean:

a. Sekuensial Pengkodean dilakukan dengan mengasosiasikan data dengan kode yangurut.

Contoh : predikat kelulusan "Sangat Memuaskan", "Cukup Memuaskan", "Memuaskan" dikodekan dengan huruf "A", "B", "C"

b. Mnemonic Pengkodean dilakukan dengan membentuk suatu singkatan dari data yang hendak dikodekan.. Contoh : "Laki-laki" -> dikodekan "L"; "Perempuan" -> dikodekan "P"

c. Blok Pengkodean dinyatakan dalam format tertentu.

Contoh : Nomor mahasiswa dengan format XX.YY.ZZZZ à terdiri atas XX = 2 digit tahun masuk, YY = 2 digit kode jurusan, ZZZZ = 4 digit nomorurut.

6.2 Merancang Tabel Relasional untuk sebuah studi kasus

Tahapan pembuatan ERD

Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.

Menentukan atribut-atribut key dari masing-masing himpunan entitas.

Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas- himpunan entitas yang ada beserta foreign key-nya.

Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi.

Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (non key).

7 Pertemuan 7

Desain Basis Data

7.1 Proses Perancangan Basis Data

a. Perancangan Sistem Database

Menurut Jogianto (1990), tujuan utama perancangan system database yaitu memiliki kemampuan menyimpan seluruh data

yang berguna dalam database. Untuk tahap desain data base secara umum yang perlu dilakukan adalah mengidentifikasi terlebih dahulu file-file yang di perlukan oleh sistem informasi.

b. Model Database

Model database menyatakan hubungan antar rekaman yang tersimpan dalam basis data. Beberapa literatur menggunakan istilah struktur data logis.

Database merupakan suatu komponen yang penting dalam sistem informasi, karena basis dalam penyediaan informasi untuk para pemakai. Penerapan database dalam sistem informasi disebut database system. Sistem database adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

7.2 Pengembangan Sistem

a. Pengembangan sistem basis data

Pengembangan basis data selalu membutuhkan kerjasama dari beberapa orang dengan keahlian yang berbeda-beda. Proses ini melibatkan pemakai, analis data, ahli komputer, database administrator, serta wakil dari pihak manajemen yang akan memakai sistem.

b. Tujuan pengembangan sistem basis data

Tujuan pengembangan sistem basis data adalah :

a. Akses data yang fleksibel (data flexibility)

o Untuk memberikan kemudahan dalam menampilkan kembali data-data yang diperlukan dan menampilkannya dalam format yang berbeda

b. Pemeliharaan Integritas data (data integrity)

o Untuk selalu meyakinkan bahwa nilai-nilai data dalam SBD adalah benar, konsisten, dan selalu tersedia

c. Proteksi data dari kerusakan dan akses ilegal (data security)

o Keamanan data diperlukan untuk melindungi data dari kerusakan yang terjadi karena alam (kebakaran, banjir, dll) atau akses yang ilegal

- o Recovery merupakan proses untuk menyusun kembali basis data yang mengalami kerusakan
 - d. Menghilangkan ketergantungan data pada program aplikasi (data independence)
 - o Ada 2 bentuk ketergantungan, yaitu logik dan fisik
 - o Ketergantungan logik, bahwa perubahan kebutuhan user terhadap data dapat berubah, tapi hal tsb tidak mengakibatkan perubahan pada pandangan user terhadap basis data
 - o Ketergantungan fisik (schema), bahwa diskripsi logik data tidak mengalami ketergantungan pada perubahan-perubahan yang terjadi dalam teknik penyimpanan secara fisik
 - e. Minimalisasi kerangkapan data (reduced data redundancy)
 - o Kerangkapan data menyebabkan media penyimpan tidak efisien, waktu akses yang lama, dan menimbulkan masalah integritas data
 - f. Penggunaan data secara bersama-sama (data shareability)
 - o SBD yang dikembangkan harus dapat digunakan oleh pemakai yang berbeda-beda
 - g. Keterhubungan data (data relatability)
 - o Adalah kemampuan untuk menetapkan hubungan logik antara tipe-tipe record yang berbeda
 - h. Standarisasi definisi rinci data (data item)
 - o Menunjukkan definisi rinci data dalam batas presisi yang digunakan pada definisi nama rinci data dan format penyimpanan dalam basis data
 - i. Meningkatkan produktivitas personal (personal productivity)
 - o SBD diharapkan mampu meningkatkan produktivitas kerja setiap personal, yang mampu memenuhi kebutuhan data sederhana hingga bentuk laporan yang lebih rumit
- c. Proses Pengembangan basis data
- Secara garis besar, proses pengembangan basis data adalah :
- Penentuan tujuan

Tujuan ditetapkan berdasar parameter pemakai dan data. Pemakai menentukan tujuan dari aplikasi yang akan dipakai. Sedangkan data menentukan bagaimana tujuan tersebut dapat dicapai.

Tujuan dinyatakan tanpa adanya kekangan, misalnya respon yang seketika, dapat dipercaya, dan perlindungan terhadap kebebasan pribadi.

- Ikatan (bindings)

Bindings merupakan ukuran tingkat fleksibilitas yang dilakukan untuk mencapai efisiensi dalam perancangan basisdata.

Ukuran-ukuran tersebut misalnya : struktur file, model basisdata, skema / relasi, pemanggilan informasi, serta perawatan data dan integritas basisdata.

Faktor fleksibilitas seringkali bertentangan dengan unjuk kerja. Jika mementingkan fleksibilitas maka struktur record menjadi sangat bermacam-macam. Jika mementingkan unjuk kerja maka akan terjadi pemaksaan pada hal-hal tertentu.

- Dokumentasi

Dokumentasi yang penting adalah model basisdata. Model basisdata akan menentukan proses yang diperlukan untuk pembentukan file, perawatan file, dan pemanggilan informasi.

Bentuk yang harus didokumentasikan adalah skema basis data, relasional basisdata, dan definisi variabel yang dipakai

- Pemrograman

Implementasi akhir setelah proses perancangan basisdata selesai adalah dengan melakukan pemrograman

- d. Langkah-langkah pengembangan sistem basis data
Komponen yang terlibat dalam pengembangan SDB : File Basis data, Software, Hardware, Personil yang terlibat
Langkah-langkah dalam pengembangan SDB :
 - a. Spesifikasi kebutuhan
 - o Definisi masalah dan studi kelayakan
 - o Rinci spesifikasi

- b. Evaluasi alternatif
 - o Indikasi alternatif
 - o Seleksi alternatif
- c. Desain
 - o Spesifikasi dan order perangkat keras
 - o Desain logik program
 - o Desain struktur data
 - o Desain prosedur untuk pemakai dan operator
 - o Definisi struktur organisasi pemakai
- d. Implementasi
 - o Instalasi dan pengujian perangkat keras
 - o Coding dan pengujian unit-unit program
 - o Konversi data
 - o Pembuatan dokumen prosedur
 - o Pelatihan pemakai
 - o Pengujian menyeluruh

- e. Langkah-langkah mendisain basis data untuk SIM
 1. Menetapkan disain / model SIM yang digambarkan dalam diagram arus data (DAD)
 2. Menentukan kebutuhan file basis data
 3. Menentukan parameter dari file basis data, meliputi :
 - a. Tipe file : file induk, file transaksi, dll.
 - b. Media file : harddisk, disket, dll
 - c. Organisasi file :
 - i. file tradisional (fileurut,urut berindeks, atau file akses langsung)
 - ii. organisasi database (struktur berjenjang, jaringan atau hubungan)
 - d. Field kunci dari file

- 4. Alat bantu dan metode dalam pengembangan sistem basis data

Alat bantu merupakan teknik yang digunakan untuk mempermudah atau mendukung kelancaran pelaksanaan kegiatan proyek

Beberapa metode :

Studi kelayakan (feasibility study)

Analisis biaya manfaat (cost benefit analysis)

Beberapa tools :

PERT (Program Evaluation and Review Technique)

Digunakan untuk penjadwalan dan pengawasan pekerjaan yang kompleks dan mempunyai sifat peka waktu, dan belum diketahui waktunya secara pasti

CPM (Critical Path Method)

Digunakan untuk mengawasi dan mengendalikan tugas-tugas dalam proyek yang telah ditentukan waktunya, dengan cara menambah atau mengurangi sumber-sumber yang diperlukan dan tersedia untuk menyelesaikan proyek

EasyCase

Digunakan sebagai alat bantu pada tahap perancangan basis data

S-Designor

Digunakan sebagai alat bantu pada tahap perancangan basis data

8 Pertemuan 8

UTS

9 Pertemuan 10,11,12

9.1 Normalisasi Data

Normalisasi dilakukan sebagai uji coba pada suatu relasi secara berkelanjutan untuk menentukan apakah relasi itu sudah baik, yaitu dapat dilakukan proses insert,update,delete, dan modifikasi pada satu atau beberapa atribut tanpa mempengaruhi integritas data dalam relasi tersebut.

9.2 Pengertian Normalisasi

Normalisasi merupakan teknik analisis data yang mengorganisasikan atribut-atribut data dengan cara mengelompokkan sehingga terbentuk entitas yang non-redundant, stabil, dan fleksible

9.3 Anomali

Anomali adalah proses pada basis data yang memberikan efek samping yang tidak diharapkan (misalnya menyebabkan ketidakonsistenan data atau membuat suatu data menjadi hilang ketika data dihapus)

Macam Anomali terdiri dari:

- Anomali peremajaan,
- Anomali Penghapusan, dan
- Anomali penyisipan

1. Anomali peremajaan

Anomali ini terjadi bila ada perubahan pada sejumlah data yang mubazir, tetapi tidak seluruhnya diubah. Contoh : Tabel Pesanan

Pemasok	Kota	Barang	Jumlah
Kartika	Jakarta	Mouse	5
Citra	Bandung	Monitor	2
Yudi	Medan	CPU	2
Citra	Bandung	Printer	1

Seandainya Citra dengan kota Bandung pindah ke Bogor maka pengubahan data hanya dilakukan pada data pertama menjadi : Tabel Pesanan

Pemasok	Kota	Barang	Jumlah
Kartika	Jakarta	Mouse	5
Citra	Bogor	Monitor	2
Yudi	Medan	CPU	2
Citra	Bandung	Printer	1

Di sini terlihat bahwa data tentang pemasok Citra tidak sama yang menyebabkan ketidakkonsistenan data.

2. Anomali Penyisipan

Anomali ini terjadi pada saat penambahan data ternyata ada elemen yang kosong dan elemen tsb justru menjadi key. Contoh : Tabel Kursus

NoSiswa	Kursus	Biaya
10	Bhs.Ingggris	60000
10	Bhs.Perancis	80000
10	Bhs.Jepang	70000
15	Bhs.Ingggris	60000
20	Bhs.Jepang	70000

Misalnya akan dibuka kursus baru yaitu Bhs.Jerman dengan biaya 75000 akan tetapi belum ada seorangpun yang ikut kursus ini, shg data menjadi : Tabel Kursus

NoSiswa	Kursus	Biaya
10	Bhs.Inggris	60000
10	Bhs.Perancis	80000
10	Bhs.Jepang	70000
15	Bhs.Inggris	60000
20	Bhs.Jepang	70000
	Bhs.Jerman	75000

3. Anomali penghapusan

Anomali ini terjadi apabila dalam satu baris/ tuple ada data yang akan dihapus sehingga akibatnya terdapat data lain yang hilang. Contoh pada table kursus data NoSiswa 20 akan dihapus karena sudah tidak ikut kursus lagi sehingga akibatnya data kursus bhs jepang dan biaya 70000 akan ikut terhapus.

9.4 Dependensi

a. Dependensi (Ketergantungan)

Konsep dasar pada tahap normalisasi yang menjelaskan hubungan atribut atau secara lebih khusus menjelaskan nilai suatu atribut yang menentukan atribut lainnya.

b. Macam-macam dependensi, yaitu :

- Dependensi fungsional

Definisi : Suatu atribut Y mempunyai dependensi fungsional terhadap atribut X jika dan hanya jika setiap nilai X berhubungan dengan sebuah nilai Y.

Notasi : $X \rightarrow Y$ (X secara fungsional menentukan Y)

Contoh : Tabel Pesanan

Pembeli	Kota	Barang	Jumlah
P1	Yogya	B1	10
P1	Yogya	B2	5
P2	Jakarta	B1	4
P2	Jakarta	B2	7
P3	Solo	B3	6
P3	Solo	B4	6

Pembeli secara fungsional menentukan kota, sebab setiap pembeli yang sama mempunyai kota yang sama, dengan demikian : Pembeli \rightarrow Kota

contoh lain : {Pembeli, Barang} \rightarrow Jumlah

Keterangan:

- Bagian yang terletak disebelah kiri tanda panah biasa disebut DETERMINAN / PENENTU dan bagian yang terletak di sebelah kanan panah disebut DEPENDENSI / YANG TERGANTUNG.
- Tanda { } biasanya digunakan untuk menentukan lebih dari satu atribut sebagai penentu atau sebagai yang tergantung.

- Dependensi fungsional sepenuhnya

Definisi : Suatu atribut Y mempunyai dependensi fungsional penuh terhadap X jika

- Y mempunyai dependensi fungsional terhadap X dan/atau
- Y tidak memiliki dependensi terhadap bagian dari X

Contoh : Pembeli → Kota

{Pembeli, Barang} → Jumlah

Intinya : Kota mempunyai dependensi fungsional terhadap Pembeli atau {Pembeli, Barang} tapi kota mempunyai dependensi fungsional sepenuhnya terhadap pembeli bukan barang.

- Dependensi Total

Definisi : Suatu atribut Y mempunyai dependensi total terhadap atribut X jika

- Y memiliki dependensi fungsional terhadap X dan
- X memiliki dependensi fungsional terhadap Y

Notasi : $X \leftrightarrow Y$

Contoh : Tabel Pemasok

KodePemasok	NamaPemasok	Kota
K1	Kartika	Jakarta
C1	Citra	Bandung
C2	Candra	Jakarta

Pada kasus ini KodePemasok \leftrightarrow NamaPemasok, karena setiap kode tidak mempunyai nama yang sama.

- Dependensi Transitif

Definisi : Atribut Z mempunyai dependensi transitif terhadap X bila :

- Y memiliki dependensi fungsional terhadap X
- Z memiliki dependensi fungsional terhadap Y

Contoh :

Kuliah	Ruang	Tempat	Waktu
Jarkom	Merbabu	Gedung Utara	Senin
Basis Data	Arjuna	Gedung Selatan	Selasa
Matematika	Merapi	Gedung Barat	Rabu
Fisika	Merbabu	Gedung Timur	Kamis

Relasi :

- Kuliah \rightarrow {Ruang, Waktu}
- Ruang \rightarrow Tempat

Terlihat bahwa : Kuliah \rightarrow Ruang \rightarrow Tempat

Dengan demikian Tempat mempunyai dependensi transitif terhadap kuliah

9.5 Diagram Dependensi Fungsional

Diagram Dependensi Fungsional (Diagram DF) Adalah diagram yang digunakan untuk menggabarkan dependensi fungsional.

Diagram ini menunjukkan hubungan antara atribut yang menjadi

penentu atribut lainnya, dengan hubungan yang dinyatakan dengan tanda panah. Seperti contoh diatas dapat digambarkan diagram DF sebagai berikut :

9.6 Dekomposisi

Pada tahap normalisasi sering kali terjadi pemecahan table kedalam bentuk dua atau lebih relasi. Proses pemecahan ini disebut dengan dekomposisi. Syarat : Tidak ada informasi yang hilang ketika suatu relasi dipecah menjadi relasi-relasi lain.

Contoh : Terdapat suatu relasi awal sebagai berikut :

Nim	Nama	Program Studi
95001	Andi	Ekonomi
95002	Vira	Teknik
95003	Andi	Fisika

Akan dibentuk kedalam dekomposisi tak hilang menjadi :

Nim	Nama	Nim	Program Studi
95001	Andi	95001	Ekonomi
95002	Vira	95002	Teknik
95003	Andi	95003	Fisika

Pada relasi awal dapat diketahui informasi sebagai berikut : 95001 adalah ANDI program studi Ekonomi. Setelah proses dekomposisi tak hilang hasilnya adalah sama 95001 adalah ANDI dan 95001 program studi Ekonomi. Contoh dekomposisi hilang adalah

Nim	Nama		Nama	Program Studi
95001	Andi		Andi	Ekonomi
95002	Vira		Vira	Teknik
95003	Andi		Andi	Fisika

95001 bernama ANDI, tetapi ANDI dengan program studi Ekonomi atau Fisika?

10 Pertemuan 13

SQL

10.1 Apakah SQL Itu ?

Structured Query Language (SQL) adalah sekumpulan perintah khusus yang digunakan untuk mengakses data dalam database relasional. SQL merupakan sebuah bahasa komputer yang mengikuti standar ANSI (American Nasional Standard Institute) yang digunakan dalam manajemen database relasional. Dengan SQL, kita dapat mengakses database, menjalankan query untuk mengambil data dari database, menambahkan data ke database, menghapus data di dalam database, dan mengubah data di dalam database. Saat ini hampir semua server database yang ada mendukung SQL untuk melakukan manajemen datanya.

10.2 SQL sebagai Sub Bahasa

Data Definition Language (DDL)

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam DDL, yaitu CREATE, ALTER, DROP.

CREATE: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru view baru, dan kolom baru.

Contoh: CREATE DATABASE nama_database

ALTER: perintah ALTER berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom.

Contoh: ALTER TABLE nama_tabel ADD nama_kolom datatype

DROP: perintah DROP berfungsi untuk menghapus database atau tabel.

Contoh: DROP DATABASE nama_database

Data Manipulation Language (DML)

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Terdapat empat perintah penting dalam DML, yaitu INSERT, SELECT, UPDATE, dan DELETE.

INSERT: perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika database dan tabel sudah dibuat.

Contoh: INSERT INTO nama_tabel VALUES (data1, data2, dst...);

SELECT: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi.

Contoh: SELECT nama_kolom1, nama_kolom2 FROM nama_tabel;

UPDATE: perintah update digunakan untuk memperbaharui data pada sebuah tabel.

Contoh: UPDATE nama_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;

DELETE: perintah delete digunakan untuk menghapus data dari sebuah tabel.

Contoh: DELETE FROM nama_tabel WHERE kolom=data;

Data Control Language (DCL)

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (priviledges). Yang termasuk perintah dalam DCL ada dua, yaitu GRANT dan REVOKE.

GRANT: perintah ini digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna. Hak akses tersebut bisa berupa hak membuat (CREATE), mengambil data (SELECT), menghapus data (DELETE), mengubah data (UPDATE), dan hak khusus lainnya yang berhubungan dengan sistem database.

REVOKE: perintah ini digunakan untuk mencabut hak akses yang telah diberikan kepada user. Dalam ini merupakan kebalikan dari perintah GRANT.

1. Antarmuka SQL terhadap DBMS

DBMS (DataBase Management System) adalah sistem yang secara khusus dibuat untuk memudahkan pemakai dalam mengelola basis data. Definisi lainnya, Database Management System adalah software yang mengijinkan penggunaanya untuk membuat, mengakses, dan mengatur sebuah database. Pada pendekatan database, setiap file yang ada pada tiap departemen disimpan pada sebuah server database dengan sebutan baru, yaitu table. Lalu setiap program dapat mengakses bagian dari database sesuai kebutuhan.

Tujuan Perancangan Database

- Untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan user secara khusus dan aplikasi – aplikasinya.
- Memudahkan pengertian struktur informasi.
- Mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan (response time, processing time, dan storage space).

Komponen DBMS

• Data Dictionary

Sebuah repository yang menyimpan data definition dan deskripsi dari struktur data didalam database

• DBMS Utilities

Program yang memungkinkan user mengelola data dengan cara create, edit, delete data dan file. Didalamnya termasuk data recovery dan back up

• Report Generator

Program untuk menghasilkan laporan yang berasal dari data yang disimpan

Faktor Pemilihan DBMS

Pemilihan DBMS didasarkan pada beberapa faktor, yakni beberapa hal teknis, ekonomi dan kebijakan organisasi.

Faktor Teknis dan Non-Teknis

Faktor teknis berhubungan dengan ketepatan DBMS yang dipilih. Yang termasuk faktor teknis adalah tipe DBMS (relational, network, hierarchical, object-relational, object lainnya), struktur penyimpanan dan jalur akses yang didukung DBMS, ketersediaan antar muka pemakai dan pemrogram, tipe bahasa query tingkat tinggi, ketersediaan alat bantu pengembangan, kemampuan berhubungan dengan DBMS lain melalui media standard, pilihan arsitektur yang berhubungan dengan operator client-server dan lain sebagainya.

Faktor non teknis termasuk di dalamnya status finansial dan dukungan organisasi terhadap vendor.

Faktor Ekonomi dan Organisasi

Hal-hal yang harus dipertimbangkan secara ekonomi dan faktor organisasi adalah :

1. Software acquisition cost : Merupakan harga "up-front" dalam pembelian perangkat lunak, termasuk pilihan bahasa, pilihan antar muka seperti form, menu dan antar muka Web berbasis GUI, pilihan recovery/backup, metode akses khusus dan dokumentasi. Versi DBMS yang tepat untuk sistem operasi harus dipilih. Biasanya, alat bantu pengembangan, alat bantu desain dan dukungan bahasa tambahan tidak termasuk dalam harga dasar.
2. Maintenance cost : Berhubungan dengan harga layanan pemeliharaan standard dari vendor dan untuk menjaga versi DBMS tetap up to date.
3. Hardware acquisition cost : perangkat keras baru mungkin diperlukan, seperti memory, terminal, disk drive dan controller baru, atau penyimpan DBMS khusus.
4. Database creation and conversion cost : Berhubungan dengan biaya pembuatan sistem basis data dari konversi sistem yang sudah ada ke perangkat lunak DBMS baru. Operasi sistem yang sudah ada dilakukan paralel dengan sistem baru sampai semua aplikasi diimplementasikan penuh dan diujicoba.

5. Personal cost : Akuisisi perangkat lunak DBMS untuk pertama kali oleh organisasi biasanya dilakukan dengan reorganisasi departemen data processing.

6. Training cost : Karena DBMS biasanya berupa sistem kompleks, personal harus ditraining menggunakan dan memprogram DBMS. Training diperlukan pada semua level, termasuk programming, pengembangan aplikasi dan administrasi basis data.

7. Operating cost : Biaya operasi lanjutan dari sistem basis data biasanya tidak termasuk dalam evaluasi.

Keuntungan DBMS tidak mudah diukur dan dihitung. DBMS mempunyai beberapa keuntungan dibandingkan sistem file, seperti mudah penggunaan, konsolidasi informasi perusahaan yang lebih luas, ketersediaan data yang lebih luas, dan akses yang lebih cepat ke informasi. Dengan akses berbasis Web, bagian data dapat dibuat akses global seperti pemakai luar. Keuntungan lainnya adalah mengurangi biaya pengembangan aplikasi, mengurangi redundancy data dan keamanan dan kontrol yang lebih baik.

2. Elemen SQL

Elemen dasar SQL mencakup pernyataan, nama, tipe data, konstanta, ekspresi, dan fungsi bawaan.

1. Pernyataan

Pernyataan adalah perintah SQL yang meminta sesuatu tindakan kepada DBMS. SQL memiliki kira-kira 30 pernyataan. Beberapa pernyataan dasar SQL dapat dilihat pada tabel berikut :

Pernyataan	Keterangan
ALTER	Mengubah struktur tabel
COMMIT	Mengakhiri sebuah eksekusi transaksi
CREATE	Menciptakan tabel, indeks atau pandangan
DELETE	Menghapus baris pada tabel
DROP	Menghapus tabel, indeks atau pandangan
GRANT	Menugaskan hak terhadap basis data kepada pengguna atau grup pengguna
INSERT	Menambahkan sebuah baris pada tabel
REVOKE	Membatalkan hak terhadap basis data

ROLLBACK Mengembalikan ke keadaan semula sekiranya suatu transaksi gagal dilaksanakan
SELECT Memilih baris dan kolom pada tabel
UPDATE Mengubah nilai pada sebuah baris

2. Nama

Nama digunakan sebagai identitas bagi objek-objek pada DBMS. Contoh objek pada DBMS adalah tabel, kolom dan pengguna.

3. Tipe Data

Setiap data memiliki tipe data. Berikut ini adalah tipe data dalam MySQL :

Tipe data untuk numerik :

Tipe Keterangan Range Nilai

TINYINT Nilai integer yang sangat kecil Signed : -128 s.d. 127 Unsigned : 0 s.d. 255

SMALLINT Nilai integer yang kecil Signed : -32768 s.d. 32767 Unsigned : 0 s.d. 65535

MEDIUMINT Integer dengan nilai medium Signed : -8388608 s.d. 8388607 Unsigned : 0 s.d. 16777215

Tipe Keterangan Range Nilai

INT Integer dengan nilai standar Signed : -2147483648 s.d. 2147483647 Unsigned : 0 s.d. 4294967295

BIGINT Integer dengan nilai besar Signed : -9223372036854775808 s.d. 9223372036854775807 Unsigned : 0 s.d. 18446744073709551615

FLOAT Bilangan desimal dengan single-precision minimum $\pm 1.175494351e-38$ maksimum $\pm 3.402823466e+38$

DOUBLE Bilangan desimal dengan double-precision minimum $\pm 2.2205738585072014e-308$ maksimum $\pm 1.7976931348623457e+308$

DECIMAL(M,D) Bilangan float (desimal) yang dinyatakan sebagai string. M adalah jumlah digit yang disimpan dalam suatu kolom, N adalah jumlah digit dibelakang koma Tergantung pada nilai M dan D

Keterangan : Signed dan Unsigned adalah atribut untuk tipe data numerik

- Signed : data yang disimpan dalam suatu kolom dapat berupa data negatif dan positif.
- Unsigned : digunakan agar data yang dimasukkan bukan data negatif (≥ 0). Tipe data float tidak dapat dinyatakan dengan unsigned.

Tipe data string :

Tipe Keterangan Ukuran Maksimum

CHAR(n) String karakter dengan panjang yang tetap, yaitu n 1 M byte

VARCHAR(n) String karakter dengan panjang yang tidak tetap, maksimum n. 1 M byte

TINYBLOB BLOB (Binary Large Object) yang sangat kecil 28-1 byte

BLOB BLOB berukuran kecil 216-1 byte

MEDIUMBLOB BLOB berukuran sedang 224-1 byte

LOB BLOB berukuran besar 232-1 byte

TINYTEXT String teks yang sangat kecil 28-1 byte

TEXT String teks berukuran kecil 216-1 byte

MEDIUMTEXT String teks berukuran medium(sedang) 224-1 byte

LONGTEXT String teks berukuran besar 232-1 byte

ENUM Enumerasi, kolom dapat diisi dengan satu member enumerasi 65535 anggota

SET Himpunan, kolom dapat diisi dengan beberapa nilai anggota himpunan 64 anggota himpunan

Tipe data tanggal dan jam :

Tipe Range Format

DATE "1000-01-01" s.d. "9999-12-31" "0000-00-00"

TIME "-832:59:59" s.d. "838:59:59" "00:00:00"

DATETIME "1000-01-01 00:00:00" s.d. "9999-12-31 23:59:59" "0000-00-00 00:00:00"

4. Konstanta

Konstanta menyatakan nilai yang tetap.

5. Ekspresi

Ekspresi adalah segala sesuatu yang menghasilkan nilai.

Ekspresi digunakan untuk menghitung nilai.

Contoh : harga*jumlah+2

Simbol-simbol yang dapat digunakan pada ekspresi aritmatika.

Simbol Keterangan

* Perkalian

/ Pembagian

+ Penjumlahan

– Pengurangan

1. Aggregate Functions (Fungsi Agregat)

Fungsi adalah sebuah subprogram yang menghasilkan suatu nilai jika dipanggil. Fungsi agregat adalah fungsi standar di dalam SQL, suatu fungsi yang digunakan untuk melakukan summary, fungsi statistik standar yang dikenakan pada suatu tabel atau query.

2. AVG(ekspresi)

Fungsi ini digunakan untuk mencari rata-rata nilai dalam suatu kolom dari suatu tabel atau ekspresi. Ekspresi dalam fungsi AVG umumnya adalah nama kolom. Kolom yang dicari nilai rata-ratanya adalah kolom dengan tipe data numerik.

3. COUNT(x)

Fungsi ini digunakan untuk menghitung jumlah record (baris) dari suatu kolom dari suatu tabel. X adalah nama kolom yang ingin dicari jumlah barisnya.

4. MAX(ekspresi)

Fungsi ini digunakan untuk mencari nilai terbesar dari suatu kolom dari suatu tabel. Kolom yang dicari nilai terbesarnya memiliki tipe data numerik.

5. MIN(ekspresi)

Fungsi ini digunakan untuk mencari nilai terkecil dari suatu kolom dari suatu tabel. Kolom yang dicari nilai terkecilnya memiliki tipe data numerik.

6. SUM(ekspresi)

Fungsi ini digunakan untuk mendapatkan nilai total dari suatu kolom pada suatu tabel.

11 Pertemuan 14

11.1 Pemrosesan Query

Query Language merupakan bahasa khusus yang digunakan untuk melakukan perintah-perintah untuk mengakses data pada sistem basis data.

Query Language atau yang lebih dikenal dengan nama SQL merupakan suatu bahasa yang digunakan untuk membuat sebuah database.

Langkah dasar pemrosesan query :

1. Parsing dan translasi

a. Translasi query menjadi bentuk internal kemudian translasikan ke aljabar relasional

b. Parser memeriksa sintaks dan verifikasi relasi-relasi

2. Optimasi : memilih evaluasi plan yang paling rendah costnya

3. Evaluasi : query-execution engine memakai query-evaluation plan, mengeksekusi plan tersebut, dan mengembalikan hasilnya
Cost umumnya diukur sebagai total waktu diperlukan untuk menyelesaikan query

1. Banyak faktor yang menentukan cost : Kecepatan akses disk, CPU, atau sampai komunikasi jaringan

2. Kecepatan akses disk adalah faktor utama penentu cost dan relatif mudah diperkirakan. Kecepatan ini diukur dengan :

a. $\text{Number of seeks} * \text{average-seek-cost}$

b. $\text{Number of blocks read} * \text{average-block-read-cost}$

c. $\text{Number of blocks written} * \text{average-block-write-cost}$

Cost untuk menulis sebuah blok lebih besar dari cost untuk membaca blok karena setelah menulis, data akan dibaca lagi untuk memastikan data yang ditulis berhasil.

Operasi-operasi dalam pemrosesan query :

1. Selection dengan menggunakan algoritma searching seperti A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11

2. Sorting

Untuk relasi yang muat di memori, teknik ini seperti quicksort, sedangkan untuk relasi yang tidak cocok pada memori, external sort-merge cukup bagus.

3. Join diimplementasikan dengan beberapa algoritma seperti nested-loop join, block nested-loop join, indexed nested-loop join, merge-join, hash-join

4. Duplicate elimination dapat diimplementasikan dengan hashing atau sorting
5. Agregation dapat diimplementasikan dengan cara yang sama pada duplicate elimination
6. Projection diimplementasikan dengan melakukan proyeksi pada setiap tupel yang diikuti dengan duplicate elimination
7. Set operations ($\wedge, \vee, -$) dapat digunakan sebagai varian dari merge-join setelah sorting atau varian dari hash-join
8. Outer join adalah modifikasi dari join dengan diikuti tambahan NULL sesuai dengan ruas yang diinginkan

1. Struktur Dasar
2. Fungsi Agregasi
3. Nilai Null
4. Manipulasi Data

Bekerja dengan SQL

1. Create Menciptakan tabel, indeks atau pandangan
2. Drop Menghapus tabel, indeks atau pandangan
3. Alter Mengubah struktur tabel
4. Insert Menambahkan sebuah baris pada tabel
5. Update Mengubah nilai pada sebuah baris
6. Delete Menghapus baris pada tabel

12 Pertemuan 15

12.1 Proteksi Data dan aplikasi Basis Data

Keamanan database adalah suatu cara untuk melindungi database dari ancaman, baik dalam bentuk kesengajaan atau pun bukan. Ancaman adalah segala situasi atau kejadian baik secara sengaja maupun tidak yang bersifat merugikan dan mempengaruhi system serta secara konsekuensi terhadap perusahaan/organisasi yang memiliki system database. Keamanan database tidak hanya berkenaan dengan data yang ada pada database saja, tetapi juga meliputi bagian lain dari system database, yang tentunya dapat mempengaruhi database tersebut. Hal ini berarti keamanan database mencakup perangkat keras, perangkat lunak, orang dan data.

Agar memiliki suatu keamanan yang efektif dibutuhkan kontrol yang tepat. Seseorang yang mempunyai hak untuk mengontrol dan mengatur database biasanya disebut Administrator database. Seorang administratorlah yang memegang peranan penting pada

suatu system database, oleh karena itu administrator harus mempunyai kemampuan dan pengetahuan yang cukup agar dapat mengatur suatu system database. Keamanan merupakan suatu proteksi terhadap pengrusakan data dan pemakaian data oleh pemakai yang tidak punya kewenangan.

System yang aman memastikan kerahasiaan data yang terdapat didalamnya. Beberapa aspek keamanan yaitu :

- Mambatasi akses ke data dan servis
- Melakukan autentifikasi pada user
- Memonitor aktivitas-aktivitas yang mencurigakan

Keamanan database dapat dikelompokan sebagai berikut :

- Pencurian dan penipuan.

Pencurian dan penipuan database tidak hanya mempengaruhi lingkungan database

tetapi juga seluruh perusahaan/organisasi. Keadaan ini dilakukan oleh orang, dimana seseorang ingin melakukan pencurian data atau manipulasi data, seperti saldo rekening, transaksi, transfer dan lain-lain. Untuk itu fokus harus dilakukan pada kekuatan system agar menghindari akses oleh orang yang tidak memiliki kewenangan.

- Hilangnya kerahasiaan dan privasi

Suatu data dapat memiliki nilai kerahasiaan, karena data tersebut merupakan

sumber daya yang strategis pada perusahaan, maka pada kasus ini data tersebut

harus diamankan dengan memberikan hak akses pada orang tertentu saja.

- Hilangnya integritas

Integritas ini berkaitan dengan akurasi dan kebenaran data dalam database, seperti

data korup. Hal ini akan secara serius mempengaruhi perusahaan/organisasi.

- Hilangnya ketersediaan

Hilangnya ketersediaan berarti data, system, keduanya tidak dapat diakses, servis mati, yang tentunya secara serius sangat mempengaruhi perusahaan/organisasi. Saat ini banyak perusahaan yang membutuhkan kemampuan system yang aktif 7 x 24 , 7 hari 1 minggu.

Berdasarkan pengelompokan tersebut, tentunya banyak aspek yang harus kita perhatikan demi terciptanya keamanan database. Bisa saja seseorang mencuri computer kita yang berisi data penting, mungkin juga karyawan yang diberi hak untuk mengakses data melakukan kejahatan dengan menjual informasi tersebut pada pihak lain demi kepentingan pribadi. Hal-hal tersebut memang termasuk kendala keamanan database yang harus mendapat perhatian, tetapi seorang administrator tidak dapat mengawasi kelemahan tersebut. Seorang administrator hanya fokus pada sistem database itu sendiri, dan hal inilah yang akan kita bicarakan. Tentunya perkembangan teknologi mengharuskan suatu perusahaan untuk mengimplementasikan system database yang bukan hanya aman tetapi juga mudah diakses dan handal, menyala 7x24 jam, 7 hari 1 minggu tanpa off.

Penyebaran informasi secara global sangat menguntungkan semua pihak. Dengan adanya internet, komunikasi antar cabang, perusahaan, konsumen dan sebagainya semakin mudah. Pemberian informasi mengenai perusahaan kepada masyarakat melalui internet merupakan salah satu strategi komunikasi, marketing, public relation perusahaan tersebut, adanya transaksi on line yang meningkatkan gaya hidup masyarakat dan lainlain. Semua itu tidak terlepas dari suatu perkembangan system database dan tentunya membuat keamanan menjadi rentan. Sangatlah mudah dalam suatu lingkungan database diciptakan suasana yang menakutkan, tanpa kepastian dan keraguan. Sebagai seorang administrator sangat perlu memperhatikan kondisi tersebut. Tentukan resiko yang sebenarnya dan selidiki apa yang dapat dilakukan terhadap kondisi itu. Sebenarnya kebanyakan database terkonfigurasi dalam keadaan yang mudah ditembus, akan tetapi hal ini bukan berarti database tidak dapat dibuat aman sebagaimana mestinya.

12.2 Ancaman terhadap database

Serangan terhadap database

Secara garis besar keamanan database dikategorikan sbb:

□ Keamanan Server

Perlindungan Server adalah suatu proses pembatasan akses yang sebenarnya pada database dalam server itu sendiri. Menurut Blake Wiedman ini adalah suatu sisi keamanan yang sangat penting dan harus direncanakan secara hati-hati. Ide dasarnya adalah kita tidak dapat mengakses apa yang kita tidak dapat lihat, atau apakah kita ingin database server kita dapat dilihat diseluruh dunia? Database kita bukanlah suatu web server, koneksi yang tidak dikenali tidak diijinkan.

□ Trusted Ip Access

Setiap server harus dapat mengkonfigurasi alamat ip yang diperbolehkan mengakses dirinya. Kita tidak mengizinkan semua orang dapat mengakses server kita sebagaimana kita tidak mengizinkan orang lain memasuki rumah kita tanpa ijin. Jika server melayani suatu web server maka hanya alamat web server itu saja yang dapat mengakses server database tersebut. Jika server database melayani jaringan internal maka hanya alamat jaringanlah yang boleh menghubungi server. Sangat perlu diperhatikan bahwa jangan pernah menggabungkan server database web dengan server database informasi internal perusahaan anda, ini adalah suatu mental yang buruk untuk seorang admin. Trusted Ip Access merupakan server database terbatas yang hanya akan memberi respon pada Ip yang dikenali saja.

□ Koneksi Database

Saat ini semakin banyaknya aplikasi dinamis menjadi sangat menggoda untuk melakukan akses yang cepat bahkan update yang langsung tanpa autentifikasi. Jangan pernah berpikir demikian, ini hanya untuk seorang pemalas. Jika kita ingin mengizinkan pemakai dapat mengubah database melalui web page, pastikan anda memvalidasi semua masukan untuk memastikan bahwa inputan

benar, terjamin dan aman. Sebagai contoh, pastikan anda menghilangkan semua code SQL agar tidak dapat dimasukan oleh user. Jika anda seorang admin yang membutuhkan koneksi ODBC, pastikan koneksi yang digunakan unik.

□ Kontrol Akses Tabel

Kontrol akses table ini adalah salah satu bentuk keamanan database yang sering diabaikan, karena cukup sulit penerapannya. Penggunaan control akses table yang benar dibutuhkan kolaborasi antara system administrator dengan pengembang database. Hal inilah yang sulit dilakukan. Pemberian ijin user untuk mengakses informasi dapat membuat informasi terbuka kepada public. Jika seorang user mengakses informasi apakah akan dilihat menggunakan session yang sama? Atau jika table digunakan sebagai referensi system mengapa ia diberikan ijin selain hak membaca saja.

12.3 Perlindungan terhadap data yang sensitif

Penyalahgunaan Database :

1. Tidak disengaja, jenisnya :

- a. kerusakan selama proses transaksi
- b. anomali yang disebabkan oleh akses database yang konkuren
- c. anomali yang disebabkan oleh pendistribusian data pada beberapa komputer
- d. logika error yang mengancam kemampuan transaksi untuk mempertahankan konsistensi database.

2. Disengaja, jenisnya :

- a. Pengambilan data / pembacaan data oleh pihak yang tidak berwenang.
- b. Pengubahan data oleh pihak yang tidak berwenang.
- c. Penghapusan data oleh pihak yang tidak berwenang.

Tingkatan Pada Keamanan Database :

1. Fisikal : Lokasi-lokasi dimana terdapat sistem komputer haruslah aman secara fisik terhadap serangan perusak.

2. Manusia : wewenang pemakai harus dilakukan dengan berhati-hati untuk mengurangi kemungkinan adanya manipulasi oleh pemakai yang berwenang
3. Sistem Operasi : Kelemahan pada SO ini memungkinkan pengaksesan data oleh pihak tak berwenang, karena hampir seluruh jaringan sistem database menggunakan akses jarak jauh.
4. Sistem Database : Pengaturan hak pemakai yang baik.

Keamanan Data :

1. Otorisasi :

- o Pemberian Wewenang atau hak istimewa (priviledge) untuk mengakses sistem atau obyek database
- o Kendali otorisasi (=kontrol akses) dapat dibangun pada perangkat lunak dengan 2 fungsi :
 - Mengendalikan sistem atau obyek yang dapat diakses
 - Mengendalikan bagaimana pengguna menggunakannya
 - Sistem administrasi yang bertanggungjawab untuk memberikan hak akses dengan membuat account pengguna.

2. Tabel View :

Merupakan metode pembatasan bagi pengguna untuk mendapatkan model

database yang sesuai dengan kebutuhan perorangan. Metode ini dapat menyembunyikan data yang tidak digunakan atau tidak perlu dilihat oleh pengguna.

Contoh pada Database relasional, untuk pengamanan dilakukan beberapa level :

1. Relasi adalah pengguna diperbolehkan atau tidak diperbolehkan mengakses langsung suatu relasi
2. View adalah pengguna diperbolehkan atau tidak diperbolehkan mengakses data yang terapat pada view
3. Read Authorization adalah pengguna diperbolehkan membaca data, tetapi tidak dapat memodifikasi.
4. Insert Authorization adalah pengguna diperbolehkan menambah data baru, tetapi tidak dapat memodifikasi data yang sudah ada.
5. Update Authorization adalah engguna diperbolehkan memodifikasi data, tetapi tidak dapat menghapus data.

6. Delete Authorization adalah pengguna diperbolehkan menghapus data.

Untuk Modifikasi data terdapat otorisasi tambahan :

1. Index Authorization adalah pengguna diperbolehkan membuat dan menghapus index data.
2. Resource Authorization adalah pengguna diperbolehkan membuat relasi-relasi baru.
3. Alteration Authorization adalah pengguna diperbolehkan menambah/menghapus atribut suatu relasi.
4. Drop Authorization adalah pengguna diperbolehkan menghapus relasi yang sudah ada.

13 Pertemuan 16

UAS

DAFTAR PUSTAKA

Korth, H., Silberschats, A., Sudarshan, S., 1998, Database System Concept, McGraw-Hill
Ramakrishnan, R., 1998, Database Management Systems, McGraw-Hill

Kadir, A., 1999, Konsep dan Tuntunan Praktis Basis Data, Penerbit Andi
Fatansyah, 2004, Basis Data, Penerbit Informatika

Ullman, J.D., 1998, Principles of Database and Knowledge Base Systems, Computer Science Press

Date, C.J., 1990, An Introduction to Database System, Addison-Wesley

Elmasri, R., Navathe, S.B., 2000, Fundamental of Database Systems, 3rd Edition, Addison Wesley

Husni I. Pohan, 2002, SQL + Tutorial plus Studi Kasus dengan Oracle dan Sybase, Penerbit Informatika Bandung

Rebecca M. Riordan, 2005, Designing Effective Database Systems, Addison Wesley Professional