

Bab 2

Desain Basis Data

POKOK BAHASAN:

- ✓ Pendahuluan
- ✓ Aturan Sistem Informasi dalam Organisasi
- ✓ Proses Desain Basis Data

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami aturan system informasi dalam organisasi
- ✓ Memahami proses desain basis data

2.1 PENDAHULUAN

Aktifitas desain basis data menggunakan proses yang sistematis yang disebut metodologi desain, dimana target basis data diatur dengan RDBMS, ORDBMS atau ODBMS. Metodologi desain menggunakan alat Bantu seperti Designer 2000 dari Oracle, ERWin, BPWin dan Paradigm Plus oleh Platinum Technology dan lain sebagainya.

Biasanya, desain basis data kecil sekitar 20 pemakai tidak perlu sangat kompleks. Tetapi untuk ukuran medium atau basis data besar yang melayani beberapa grup aplikasi yang luas, puluhan sampai ratusan pemakain, pendekatan sistematis menjadi sangat perlu untuk melakukan desain basis data.

Basis data yang besar dengan data beberapa puluh sampai gigabyte dan skema dengan lebih dari 30 sampai 40 tipe entity yang berbeda, dapat memenuhi array yang besar dari basis data pemerintahan, industri dan institusi financial dan komersial. Sektor industri termasuk di dalamnya bank, hotel, airline, asuransi, utilitas dan komunikasi menggunakan basis data untuk operasi setiap hari 24 jam, 7 hari per minggu atau operasi 24 kali 7. Sistem aplikasi untuk basis data tersebut disebut system pemrosesan transaksi untuk volume transaksi besar.

2.2 ATURAN SISTEM INFORMASI DALAM ORGANISASI

2.2.1 Organizational Context untuk Penggunaan Sistem Basis Data

Sistem basis data menjadi bagian dari sistem informasi dari beberapa organisasi. Tahun 1960 an sistem informasi didominasi dengan sistem file, tetapi sejak awal 1970 an organisasi mulai berpindah ke sistem basis data. Untuk mengakomodasi sistem, beberapa organisasi membuat posisi administrator basis data (DBA) untuk mengontrol aktifitas basis data. Kemudian, information resource management (IRM) juga diperkenalkan oleh organisasi yang besar sebagai kunci kesuksesan manajemen bisnis. Terdapat beberapa alasan :

- Data dianggap sebagai resource yang bekerjasama, dan manajemen dan kontrol dilakukan terpusan untuk pekerjaan yang lebih efisien dalam organisasi
- Fungsi dalam organisasi dikomputerisasi, sebagai kebutuhan ketersediaan data yang besar dan up to date.
- Seiring pertumbuhan data dan aplikasi relasi yang lebih kompleks dari data perlu dimodelkan dan diatur.
- Terdapat konsolidasi dari information resource pada beberapa organisasi.

Sistem basis data memenuhi 4 kebutuhan seperti dijelaskan sebelumnya dalam ukuran besar. Dua karakteristik tambahan dari sistem basis data yang juga sangat bernilai :

- Data independence mem-proteksi program aplikasi dari perubahan dalam organisasi logika dan akses fisik dan struktur penyimpanan.

- External shemas (views) memungkinkan data yang sama digunakan untuk beberapa aplikasi dengan setiap aplikasi mempunyai pandangan sendiri terhadap data.

Sistem basis data menyediakan aplikasi baru yaitu :

- Integritas data pada multiple aplikasi ke dalam basis data tunggal.
- Pengembangan yang simple dengan menggunakan bahasa tingkat tinggi seperti SQL.
- Kemungkinan didukung untuk browsing dan query oleh manajer dalam pemrosesan transaksi level produksi yang besar.

Sejak awal 1970 sampai pertengahan 1980, perubahan pembuatan data repository tersentral yang besar dengan DBMS tunggal tersentral. Selama 10 sampai 15 tahun, pengembangan basis data meliputi :

1. Personal computer dan produk software seperti basis data, seperti EXCEL, FOXPRO, MSSQL, ACCESS atau SQL
2. DBMS terdistribusi dan client-server sebagai pembuka pilihan mendistribusikan basis data ke banyak sistem komputer untuk kontrol yang lebih baik dan proses lokal yang lebih cepat. Alat bantu pengembangan aplikasi seperti POWERBUILDER atau Developer 2000 (oleh Oracle) lebih mudah digunakan dengan fasilitas built-in untuk menghubungkan aplikasi ke server basis data.
3. Beberapa organisasi sekarang menggunakan sistem data dictionary atau information repository, yaitu DBMS min yang mengatur metadata yaitu data yang menggambarkan struktur basis data, constraints, aplikasi, otorisasi dan sebagainya. Sistem data dictionary menyimpan dan mengatur informasi berikut :
 - a. Deskripsi skema sistem basis data.
 - b. Informasi detail dari desain fisik basis data, seperti struktur penyimpanan, akses path, ukuran file dan record.
 - c. Deskripsi pemakai basis data, tanggung jawab dan hak akses.
 - d. Deskripsi tingkat tinggi dari transaksi basis data dan aplikasi dan relasi pemakai ke transaksi.

- e. Relasi antara transaksi basis data dan data item yang dirujuk. Hal ini sangat berguna untuk menentukan transaksi mana yang diakibatkan jika definisi data diubah
- f. Penggunaan statistik seperti frekuensi query dan transaksi dan jumlah akses ke basis data

2.2.2 Siklus Sistem Informasi

Pada organisasi yang besar, sistem basis data adalah bagian dari sistem informasi, di dalamnya termasuk semua resource yang dilibatkan dalam koleksi, manajemen, penggunaan dan disseminasi information resource dari organisasi. Pada sistem komputerisasi, resource adalah data itu sendiri, perangkat lunak DBMS, perangkat keras komputer sistem, media penyimpanan, personal yang menggunakan dan mengatur data (DBA, pemakai akhir, dan pemakai dsb), perangkat lunak aplikasi yang mengakses dan mengubah data dan programmer aplikasi yang mengembangkan aplikasi.

Siklus sistem informasi disebut siklus makro, dimana siklus sistem basis data dirujuk ke siklus mikro. Siklus makro meliputi beberapa tahap yaitu :

1. Feasibility analysis : tahap ini berhubungan dengan analisa area aplikasi potensial, mengidentifikasi sisi ekonomi dari information gathering and dissemination, membentuk studi keuntungan awal, menentukan kompleksitas data dan proses, mengatur prioritas aplikasi.
2. Requirement collection and analysis : Kebutuhan detail dikumpulkan dengan interaksi dengan pemakai potensial dan kelompok pemakai untuk mengidentifikasi permasalahan dan kebutuhan khusus. Ketergantungan aplikasi, komunikasi dan prosedur pelaporan diidentifikasi.
3. Desain : Tahap ini mempunyai dua aspek yaitu mendesain sistem basis data dan mendesain sistem aplikasi (program) yang menggunakan dan memproses basis data.
4. Implementasi : Sistem informasi diimplementasi, basis data dibentuk dan transaksi basis data diimplementasikan dan diujicoba.

5. Validation and acceptance testing : Tingkat akses dari sistem dalam memenuhi kebutuhan pemakai dan kriteria performansi divalidasi. Sistem diujicoba dengan kriteria performansi dan spesifikasi kelakukan.
6. Deployment, operation and maintenance : Pada tahap ini dilakukan konversi pemakai dari sistem lama ke sistem baru melalui training. Tahap operasional mulai jika semua fungsi sistem dioperasikan dan divalidasi. Jika kebutuhan baru atau aplikasi bertambah, maka harus melalui semua tahap sebelumnya sampai semua divalidasi dan berhubungan dengan sistem. Monitoring performansi sistem dan pemeliharaan sistem merupakan aktifitas yang penting selama tahap operasi.

2.2.3 Siklus Sistem Aplikasi Basis Data

Aktifitas yang berhubungan dengan siklus sistem aplikasi basis data meliputi tahap berikut :

1. System definition : Scope dari sistem basis data, pemakai dan aplikasi didefinisikan. Antarmuka untuk pemakai, batasan response time dan kebutuhan penyimpan dan pemrosesan diidentifikasi.
2. Database design : Pada akhir dari tahap ini , desain logika dan fisik dari sistem basis data dari DBMS sudah siap.
3. Database implementation : Tahap ini meliputi proses menentukan definisi basis data konseptual, eksternal dan internal, membuat file basis data kosong dan implementasi aplikasi perangkat lunak.
4. Loading or data conversion : Basis data dipopulasikan dengan menyimpan data langsung atau mengubah file yang sudah ada ke format sistem basis data.
5. Application conversion : aplikasi perangkat lunak dari sistem pendahulu dikonversikan ke sistem baru.
6. Testing and validation : sistem baru diuji coba dan divalidasi
7. Operation : sistem basis data dan aplikasi dioperasikan. Biasanya sistem lama dan baru dioperasikan secara paralel dalam beberapa waktu.
8. Monitoring and maintenance : selama tahap operasional, sistem secara tetap dimonitor dan dipelihara. Perubahan dan pengembangan dapat terjadi baik

pada isi data maupun aplikasi perangkat lunak. Modifikasi dan reorganisasi diperlukan dari waktu ke waktu.

2.3 PROSES DESAIN BASIS DATA

Sekarang kita fokuskan ke siklus sistem aplikasi basis data yaitu desain basis data. Tujuan desain basis data adalah :

- Memenuhi kebutuhan isi informasi dari pemakai dan aplikasi tertentu.
- Menyediakan struktur informasi alami dan mudah dipahami.
- Mendukung kebutuhan pemrosesan dan performansi obyektif seperti response time, processing time dan storage space.

Terdapat enam tahap utama pada proses desain basis data yaitu :

1. Koleksi dan analisa kebutuhan
2. Desain basis data konseptual
3. Pemilihan DBMS
4. Pemetaan model data (disebut juga desain basis data logika)
5. Desain basis data fisik.
6. Implementasi dan tuning sistem basis data

Proses desain terdiri dari dua aktifitas paralel seperti pada Gambar 2-1. Aktifitas pertama meliputi desain data content dan struktur basis data, kedua berhubungan dengan aplikasi basis data.

Enam tahap diatas tidak diproses secara berurutan. Pada beberapa kasus mungkin memodifikasi desain dari awal tahap selama tahap kemudian. Feedback loop antar tahap juga dalam tahap sering terjadi. Summary tahap 2, 4, dan 5 adalah berikut :

- Desain basis data konseptual (Tahap 2) : Tujuan dari tahap ini adalah memproduksi skema konseptual untuk basis data yang independen dari DBMS tertentu. Biasanya menggunakan model data tingkat tinggi seperti model ER atau EER.
- Pemetaan model data (Tahap 4) : Selama tahap ini yang juga disebut desain basis data logika, dilkakukan pemetaan skema konseptual dari model data tingkat tinggi ke model data DBMS.

2.3.1 Tahap 1 : Koleksi dan Analisa Kebutuhan

Sebelum melakukan desain basis data, harus mengetahui dan menganalisa keinginan pemakai terhadap suatu basis data sedetail mungkin. Proses ini disebut koleksi dan analisa kebutuhan. Untuk menentukan kebutuhan, pertama kali harus diidentifikasi bagian lain dari sistem informasi yang berhubungan dengan sistem basis data. Termasuk di dalamnya pemakai dan aplikasi baru dan yang sudah ada, kemudian kebutuhan dikoleksi dan dianalisa. Aktifitas yang merupakan bagian dari tahap ini adalah :

1. Area aplikasi mayor dan kelompok pemakai yang akan menggunakan basis data atau pekerjaan apa yang akan diakibatkan diidentifikasi.
2. Dokumen yang sudah ada yang berhubungan dengan aplikasi dipelajari dan dianalisa. Dokumen lain seperti police manual, form, report dan diagram organisasi di-review untuk menentukan apakah terdapat tambahan pada koleksi kebutuhan dan spesifikasi proses.
3. Lingkungan operasi saat ini dan rencana penggunaan informasi dipelajari. Termasuk di dalamnya analisa tipe transaksi dan frekuensi penggunaannya dan aliran informasi dalam sistem. Karakteristik geografi seperti pemakai, transaksi asli, tujuan pelaporan dipelajari. Data input dan output untuk transaksi ditentukan.
4. Penulisan respon untuk menentukan pertanyaan terkadang dikelompokkan dari pemakai basis data potensial atau kelompok pemakai. Pertanyaan ini melibatkan prioritas pemakai dan tempat yang penting untuk suatu aplikasi. Individu dilakukan interview untuk menolong dalam memperoleh informasi yang berharga dan setting prioritas.

Analisa kebutuhan dibawa ke user akhir atau pelanggan sistem basis data oleh tim ahli analis kebutuhan. Kebutuhan awal lebih informal, tidak lengkap, tidak konsisten dan sebagian tidak benar. Perlu pekerjaan yang lebih banyak untuk mentransformasi keebutuhan awal ke aplikasi yang lebih spesifik yang dapat digunakan oleh pengembangan sebagai langkah awal untuk menulis implementasi dan uji coba.

Untuk transformasi kebutuhan ke struktur yang lebih baik, teknik spesifikasi kebutuhan digunakan. Misalnya OOA (object-oriented analysis) dan DFD (data flow

diagram). Metode tersebut menggunakan teknik diagram untuk mengorganisasi dan menampilkan kebutuhan proses informasi. Dokumentasi tambahan dalam bentuk teks, tabel, grafik dan keputusan melengkapi diagram tersebut.

2.3.2 Tahap 2 : Desain Basis Data Konseptual

Tahap kedua dari perancangan basis data melibatkan dua aktifitas paralel. Aktifitas pertama yaitu desain skema konseptual, menentukan kebutuhan data yang dihasilkan pada tahap 1 dan menghasilkan skema basis data konseptual. Aktifitas kedua, desain transaksi dan aplikasi, menentukan analisa aplikasi basis data pada tahap 1 dan menghasilkan spesifikasi level tinggi untuk aplikasi tersebut.

Tahap 2a : Desain Skema Konseptual

Skema konseptual diproduksi dari tahap ini terdiri dari model data tingkat tinggi DBMS-independent dengan beberapa alasan :

1. Tujuan desain skema konseptual adalah skema lengkap tentang struktur basis data, semantik, interrelationship dan constraint. Hal ini tergantung dari DBMS yang digunakan.
2. Skema konseptual tidak tersedia sebagai deskripsi stabil dari isi basis data. Pemilihan DBMS dan keputusan desain dapat berubah tanpa mengubah skema konseptual DBMS-independent.
3. Skema konseptual yang baik sangat penting untuk pemakai basis data dan desainer. Penggunaan model data tingkat tinggi lebih ekspresif dan umum daripada model data dari DBMS.
4. Deskripsi diagram dari skema konseptual dapat menawarkan kendaraan komunikasi yang baik diantara pemakai basis data, desainer dan analyst. Karena model data level tinggi biasanya berbentuk konsep dan mudah untuk mengerti daripada model data DBMS yang level lebih rendah, atau definisi sintak data, komunikasi yang berhubungan dengan desain skema menjadi lebih kelihatan.

Pada tahap desain basis data, perlu menggunakan model data konseptual level tinggi dengan karakteristik :

1. Expressiveness : model data cukup ekspresif untuk membedakan perbedaan tipe data, relationship dan constraint.

2. Simplicity and understandability : model cukup sederhana untuk pemakai yang tidak mengerti dan menggunakan konsep tersebut.
3. Minimality : model mempunyai sejumlah kecil konsep dasar yang berbeda dan tidak overlapping.
4. Diagrammatic representation : model dalam bentuk notasi diagram untuk menampilkan skema konseptual yang mudah diinterpretasikan.
5. Formality : skema konseptual ditampilkan dalam model data harus merepresentasikan spesifikasi formal data. Sehingga, konsep model harus ditentukan secara akurat dan tidak berganda.

Pendekatan ke Desain Skema Konseptual

Untuk desain skema konseptual, harus diidentifikasi komponen dasar dari skema : tipe entiti, tipe relationship dan atribut. Harus juga menentukan key attributes, cardinality dan participation constraint, weak entity dan hierarki spesifcation / generatization. Terdapat dua pendekatan untuk merancang skema konseptual, yang diturunkan dari kebutuhan yang dikoleksi.

Pendekatan pertama adalah pendekatan desain skema terpusat (one-shot), dimana kebutuhan dari aplikasi yang berbeda dan kelompok pemakai pada tahap 1 digabungkan ke dalam satu himpunan kebutuhan sebelum desain skema dimulai. Suatu skema berhubungan digabungkan ke himpunan kebutuhan kemudian dilakukan desain. Jika terdapat banyak pemakai dan banyak aplikasi, penggabungan semua kebutuhan dapat menghabiskan waktu. Asumsikan DBA bertanggung jawab untuk menentukan bagaimana menggabungkan kebutuhan dan untuk merancang skema konseptual untuk keseluruhan basis data. Jika suatu skema konseptual dirancang dan final, skema eksternal untuk kelompok pemakai dan aplikasi dapat ditentukan oleh DBA

Pendekatan kedua adalah pendekatan view integration, dimana kebutuhan tidak digabungkan. Suatu skema dirancang untuk setiap kelompok user atau aplikasi berdasarkan kebutuhan masing-masing. Kemudian dikembangkan skema level tinggin (view) untuk setiap kelompok user atau aplikasi. Selama tahap view integration, skema bagian digabungkan ke dalam skema konseptual global untuk keseluruhan basis data. Individual view dapat dibentuk sebagai skema eksternal setelah view integration.

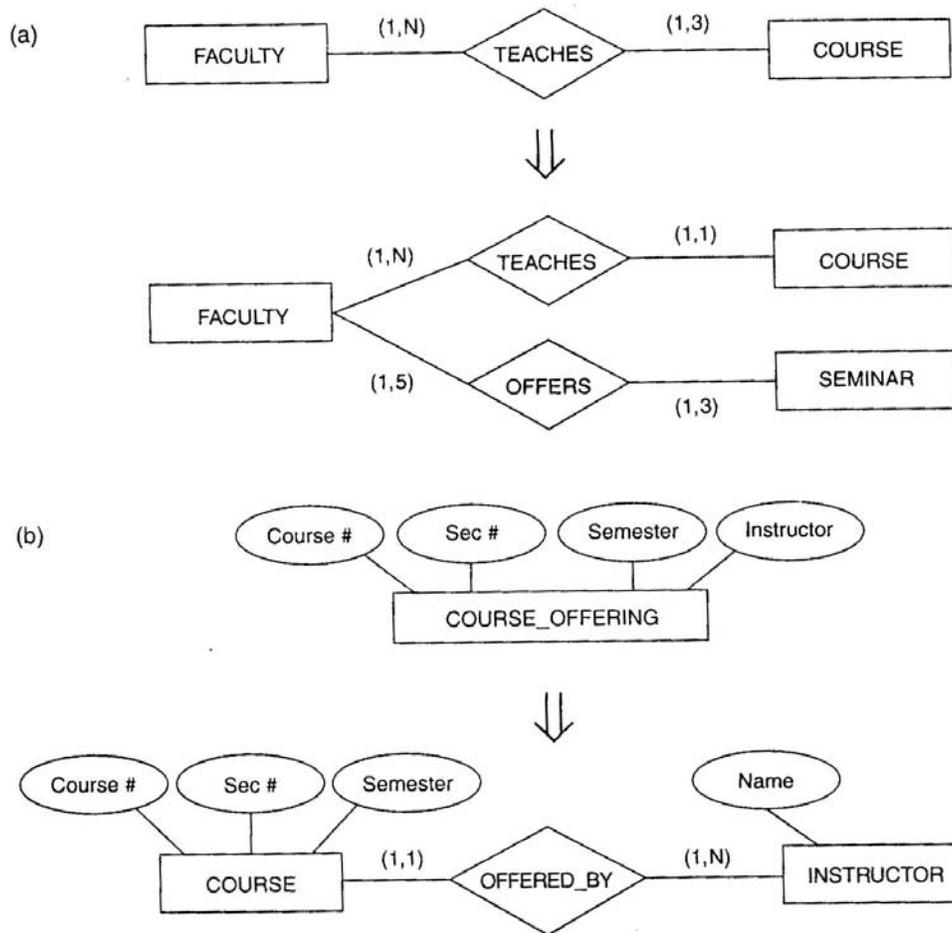
Perbedaan kedua pendekatan terletak pada tujuan dimana banyak view atau kebutuhan dari banyak pemakai dan aplikasi digabungkan. Pada pendekatan terpusat, rekonsiliasi dilakukan secara manual oleh DBA. Hal ini dapat mengakibatkan terjadi konflik pada staff DBA. Permasalahan ini dipecahkan dengan menggunakan konsultan luar.

Pada pendekatan view integration, setiap kelompok pemakai merancang skema konseptual (EER) masing-masing. Kemudian proses integrasi diaplikasikan pada skema ini (view) oleh DBA untuk membentuk skema integrasi global. Meskipun view integration dapat dilakukan manual, aplikasi ini adalah basis data besar yang melibatkan puluhan kelompok pemakai membutuhkan suatu metodologi dan penggunaan alat bantu otomatis untuk integrasi. Korespondensi antara atribut, tipe entiti dan relationship harus ditentukan sebelum integrasi dapat dilakukan.

Strategi untuk Desain Skema

Terdapat beberapa strategi untuk merancang skema, yaitu :

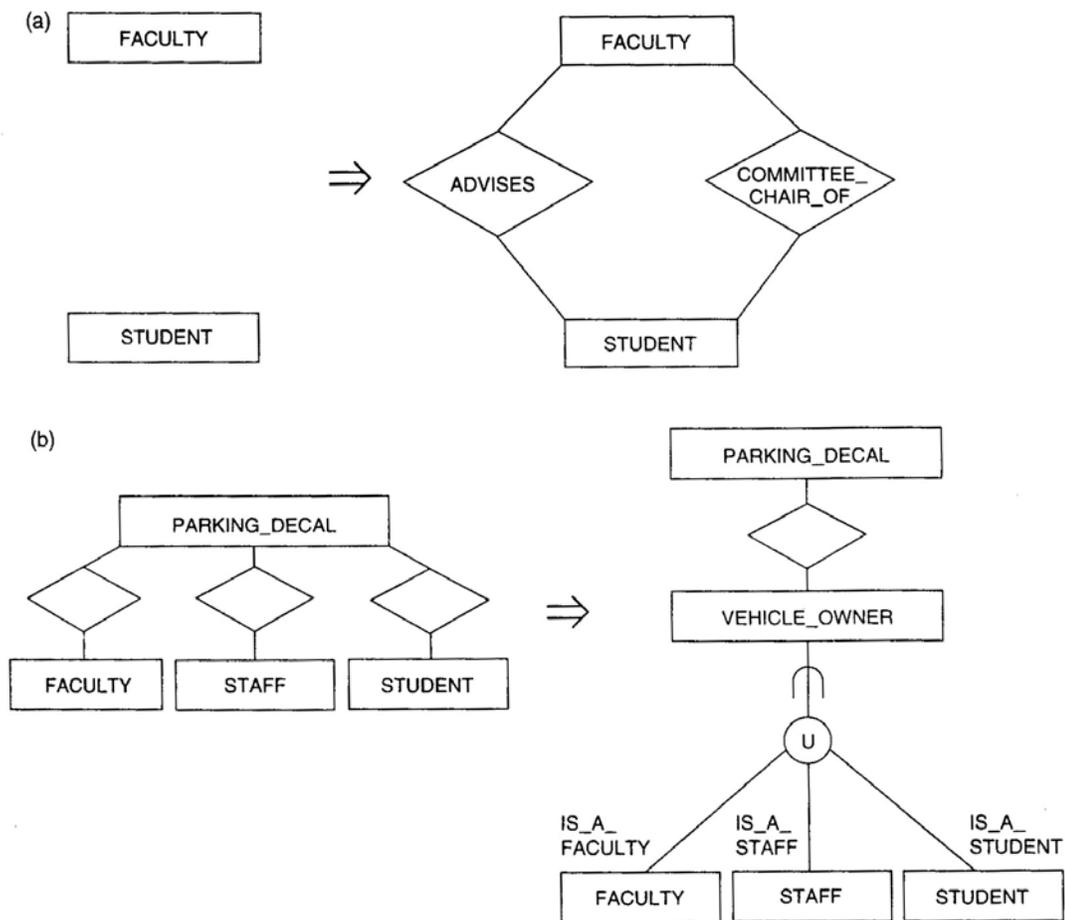
1. Top-down strategy : Dimulai dengan skema yang berisi abstraksi level tinggi dan kemudian mengaplikasikan ketentuan top-down. Sebagai contoh, tentukan hanya beberapa tipe entiti level tinggi dan kemudian lakukan pembagian ke dalam tipe entiti level lebih rendah dan relationship.
2. Bottom-up strategy : Mulai dengan skema yang berisi abstraksi dasar dan kemudian kombinasikan atau tambahkan abstraksi tersebut. Sebagai contoh, mulai dengan atribut dan kelompok ke dalam tipe entiti dan relationship. Tambahkan relasi baru pada tipe entiti selama proses perancangan.
3. Inside-out strategy : Merupakan kasus khusus dari bottom-up strategi, dimana atensi difokuskan pada himpunan konsep terpusat yang lebih nyata. Model kemudian diisi dengan konsep baru pada konsep yang sudah ada. Kita dapat tentukan beberapa tipe entiti nyata dalam skema dan dilanjutkan dengan menambah tipe entiti dan relasi yang berhubungan.
4. Mixed strategy : Kebutuhan dibagi berdasarkan top-down strategy, bagian skema dirancang untuk setiap partisi berdasarkan bottom-up strategy. Jadi strategi ini mengkombinasikan beberapa skema.



Gambar 2-2: Strategi top-down. (a) membangkitkan tipe entiti baru.
(b) dekomposisi tipe entiti ke dalam dua tipe entiti dan relasi

Gambar 2-2 dan 2-3 menggambarkan top-down strategy dan bottom-up strategy. Contoh top-down didekomposisi dari tipe entiti ke dalam beberapa tipe entiti. Gambar 2-2(a) menunjukkan COURSE dibagi dalam COURSE dan SEMINAR, dan relasi TEACHES dihubungkan terpisah dalam TEACHES dan OFFERS. Gambar 2-2(b) terlihat tipe entiti COURSE_OFFERING dibagi dalam dua tipe entiti COURSE dan INSTRUCTOR dan relasi antar keduanya. Gambar 2-3(a) memperlihatkan bottom-up strategy dari generalisasi relasi baru diantara tipe entiti. Bottom-up menggunakan kategory (tipe union) yang diilustrasikan pada Gambar 2-3(b) dimana konsep baru

VEHICLE_OWNER ditemukan dari tipe entiti yang sudah ada yaitu FACULTY, STAFF dan STUDENT.



Gambar 2-3: Contoh strategi bottom-up. (a) menemukan dan menambah relasi baru. (b) menemukan katagori baru (tipe union) dan menghubungkannya.

Skema Integrasi (View)

Untuk basis data yang besar dengan pemakai dan aplikasi yang diharapkan, pendekatan view integration untuk merancang skema individuan dan kemudian menggabungkannya. Karena individual view relatif kecil, perancangan skema lebih sederhana. Tetapi diperlukan metodologi untuk integrasi view ke skema basis data global. Skema integrasi dibagi ke dalam beberapa bagian :

1. Identifikasi korespondensi dan konflik diantara skema : Karena skema dirancang individual, perlu menentukan konstruksi khusus dalam skema yang merepresentasikan konsep dunia nyata yang sama. Korespondensi diidentifikasi sebelum integrasi diproses. Selama proses tersebut, beberapa tipe konflik diantara skema ditemukan, antara lain :
 - a. Naming conflict : Terdapat dua tipe synonym dan homonym. Synonym terjadi jika dua skema menggunakan nama yang berbeda dan menggambarkan konsep yang sama, misalnya, tipe entiti CUSTOMER pada satu skema mungkin digunakan sama dengan konsep tipe entiti CLIENT pada skema lain. Homonym terjadi jika dua skema menggunakan nama yang sama untuk menggambarkan konsep yang berbeda, sebagai contoh tipe entiti PART merepresentasikan perangkat komputer dalam skema satu dan perangkat mebel dalam skema dua.
 - b. Type conflicts : Konsep yang sama kemungkinan direpresentasikan dalam dua skema dengan konstruksi pemodelan yang berbeda. Sebagai contoh, konsep DEPARTMENT mungkin tipe entiti dalam skema satu dan atribut dalam skema lain.
 - a. Domain (value set) conflict : Suatu atribut berbeda domain dalam dua skema. Sebagai contoh, SSN dideklarasikan sebagai integer dalam skema satu dan karakter string dalam skema lain. Konflik unit pengukuran dapat terjadi jika satu skema merepresentasikan WEIGHT dalam pon dan lainnya dalam kilogram.
 - b. Konflik diantara constraint : Dua skema mungkin mempunyai constrain berbeda, sebagai contoh, key pada tipe entiti mungkin berbeda setiap skema. Contoh lain melibatkan constraint terstruktur yang berbeda pada relasi seperti TEACHES; satu skema mungkin direpresentasikan 1:N sementara lainnya M:N.
2. Modifikasi view untuk kesesuaian dengan lainnya : satu skema dimodifikasi sehingga sesuai dengan skema lainnya. Beberapa konflik diidentifikasi sebagai hal perama yang harus dipecahkan.
3. Menggabungkan view : Skema global dibuat dengan menggabungkan skema individu. Konsep yang berhubungan direpresentasikan hanya sekali dalam

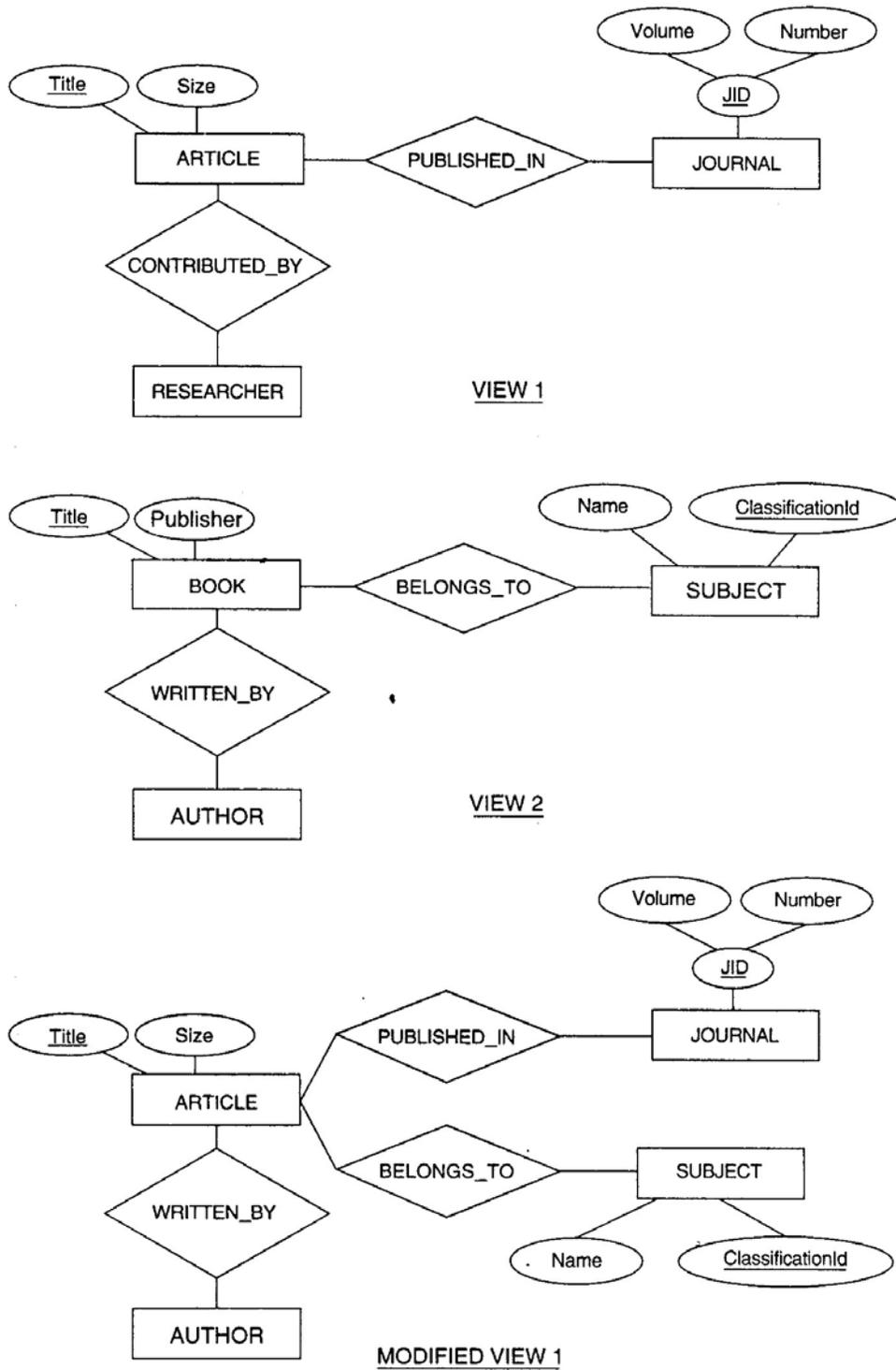
skema global dan pemetaan antara view dan skema global ditentukan. Hal ini merupakan langkah yang sulit untuk melakukan pada basis data yang sebenarnya yang melibatkan ratusan entiti dan relasi. Hal ini melibatkan sejumlah intervensi manusia dan negosiasi untuk memecahkan konflik dan mendapatkan solusi yang dapat diterima untuk skema global.

4. Restrukturisasi : sebagai langkah akhir, skema global dianalisa dan direstrukturisasi untuk memindahkan redundansi dan kompleksitas yang tidak perlu.

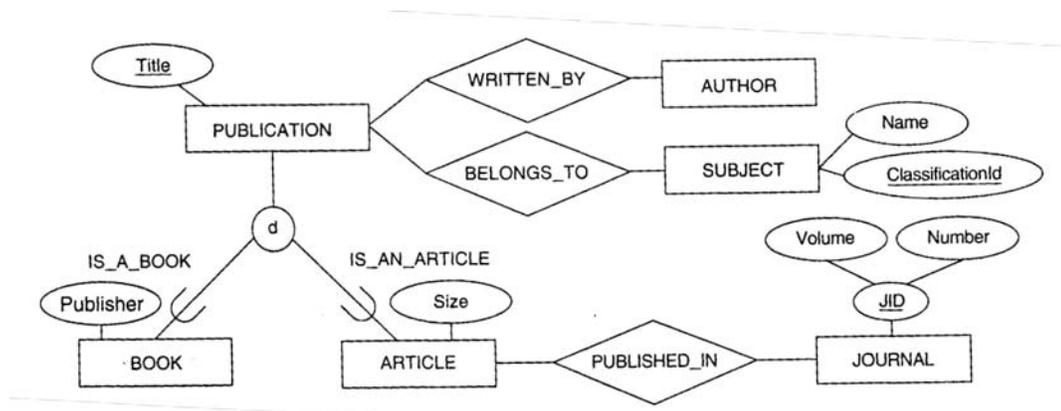
Beberapa ide diatas dapat dilihat pada contoh sederhana Gambar 2-4 dan 2-5. Gambar 2-4, dua view digabungkan untuk membuat basis data bibliografi. Selama identifikasi dari korespondensi antara dua view, ditemukan RESEARCHER dan AUTHOR adalah synonym, demikian juga CONTRIBUTED_BY dan WRITTEN_BY. Kemudian, menentukan modifikasi VIEW 1 untuk memasukan SUBJECT untuk ARTICLE, seperti Gambar 2-4, untuk konfirmasi ke VIEW 2. Gambar 2-5 menunjukkan hasil penggabungan sebagai MODIFIED VIEW 1 dengan VIEW 2. Generalisasi tipe entiti ARTICLE dan BOOK ke dalam tipe entiti PUBLICATION, dengan atribut umum Title. Relasi CONTRIBUTED_BY dan WRITTEN_BY digabungkan, demikian juga tipe entiti RESEARCHER dan AUTHOR. Atribut Publisher diaplikasikan hanya pada tipe entiti BOOK dimana atribut Size dan relasi tipe PUBLISHED_IN diaplikasikan hanya ke ARTICLE.

Tahap 2b : Desain Transaksi

Tujuan dari tahap 2b, dimana proses dilakukan paralel dengan tahap 2°, untuk mendesain karakteristik transaksi basis data yang diketahui (aplikasi) dengan cara DBMS-independent. Jika suatu sistem basis data dirancang, perancang sadar beberapa aplikasi yang diketahui (atau transaksi) yang akan dijalankan dalam basis data diimplementasikan. Bagian terpenting dari perancangan basis data adalah menentukan karakteristik fungsi transaksi tersebut sebelumnya dalam proses perancangan. Hal ini menjamin skema basis data akan memasukan semua informasi yang dibutuhkan oleh transaksi tersebut.



Gambar 2-4: Modifikasi view untuk konfirmasi sebelum integrasi.



Gambar 2-5: Skema terintegrasi setelah menggabungkan view 1 dan view 2.

Teknik yang umum digunakan untuk menentukan transaksi pada level konseptual adalah mengidentifikasi input/output dan functional behavior. Dengan menentukan parameter input dan output dan aliran fungsi internal, desainer dapat menentukan transaksi secara konseptual dan dengan cara system-independent. Transaksi dikelompokkan dalam tiga kategori :

- (1) Retrieval transaction, yang digunakan untuk menampilkan data ke layar atau untuk produksi pelaporan.
- (2) Update transaction, yang digunakan untuk memasukkan data baru atau memodifikasi data yang sudah ada pada basis data.
- (3) Mixed transaction, yaitu digunakan untuk aplikasi yang kompleks yang melakukan retrieval dan update. Sebagai contoh, misalnya basis data pemesanan tiket pesawat (airline reservation). Retrieve transaction menampilkan daftar semua pesawat pagi antara dua kota. Update transaction berupa booking tempat duduk pada jalur tertentu. Mixed transaksi pada penampilan beberapa data seperti menampilkan reservasi pelanggan pada beberapa penerbangan, dan kemudian mengubah basis data seperti membatalkan reservasi dengan menghapusnya, atau menambah segmen penerbangan untuk reservasi yang sudah ada. Transaksi (aplikasi) dapat menggunakan POWER BUILDER atau Developer 2000 (Oracle).

2.3.3 Tahap 3 : Pemilihan DBMS

Pemilihan DBMS berdasarkan beberapa faktor, beberapa hal teknis, ekonomi dan kebijakan organisasi. Faktor teknis berhubungan dengan ketepatan DBMS yang dipilih. Yang termasuk faktor teknis adalah tipe DBMS (relational, object-relational, object, lainnya), struktur penyimpanan dan akses path 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 standatd, pilihan arsitektur yang berhubungan dengan operator client-server dan lain sebagainya. Faktor non teknis termasuk di dalamnya status finansial dan dukungan organisasi terhadap vendor. 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 standart 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. Basis data sudah digunakan pada banyak organisasi, keputusan berpindah dari aplikasi berbasis file ke basis data terpusat dikarenakan faktor-faktor berikut :

1. Kompleksitas data : Relasi data menjadi lebih kompleks, memerlukan DBMS yang kuat.
2. Sharing diantara aplikasi : Semakin besar sharing antar aplikasi, semakin banyak redundansi file dan lebih besar kebutuhan akan DBMS
3. Perumbuhan dan perubahan data secara dinamis : Jika data berubah secara konstan, lebih mudah untuk melakukan perubahan dengan DBMS dibandingkan dengan sistem file.
4. Frekuensi permintaan ad hoc data : Sistem file tidak cukup tepat untuk penampilan data ad hoc
5. Volume data dan kebutuhan untuk kontrol : Volume data yang besar dan kebutuhan mengontrol memerlukan DBMS

Beberapa faktor ekonomi dan organisasi yang berakibat pemilihan suatu DBMS:

1. Organization-wide adoption of a certain philosophy : Biasanya merupakan faktor dominan yang berakibat pada penerimaan model data (misalnya, relational versus obyek), vendor, metodologi pengembangan dan alat bantu (misalnya, penggunaan analisa berorientasi obyek dan alat bantu desain dan metodologi dibutuhkan oleh semua aplikasi baru.
2. Familiarity of personnel with the system : Jika staff programming dalam organisasi familiar dengan DBMS tertentu, dapat mengurangi biaya training dan waktu pembelajaran.

3. Availability of vendor service : ketersediaan asisten vendor dalam pemecahan permasalahan dengan sistem sangat penting, karena perubahan dari non-DBMS ke lingkungan DBMS kebanyakan membutuhkan bantuan vendor pada awalnya.

Beberapa DBMS sekarang mempunyai versi yang berjalan pada beberapa konfigurasi perangkat keras / perangkat lunak (platform). Kebutuhan aplikasi untuk backup, recovery, performansi, integritas dan sekuriti harus juga dipertimbangkan. Beberapa DBMS sekarang dirancang sebagai solusi total untuk pemrosesan informasi dan manajemen sumber daya informasi yang diperlukan dalam organisasi. Kebanyakan vendor DBMS mengkombinasikan produk mereka dengan pilihan berikut :

- Editor teks dan browser
- Pembangkit laporan dan daftar utilitas
- Perangkat lunak komunikasi
- Entri data dan menampilkan form, layar, dan menu dengan pengeditan otomatis
- Alat bantu untuk mengakses World Wide Web
- Alat bantu merancang basis data grafis.

2.3.4 Tahap 4 : Pemetaan Model Data (Desain Basis Data Logika)

Tahap berikutnya dari perancangan basis data adalah membuat skema konseptual dan skema eksternal dalam model data dari DBMS terpilih dengan memetakan skema tersebut. Proses pemetaan dalam dua bentuk :

1. System-independent mapping : Pada bentuk ini, pemetaan tidak mempertimbangkan karakteristik khusus data atau kasus khusus yang diaplikasikan ke implementasi DBMS dari model data.
2. Tailoring the schemas to aspecific DBMS : DBMS yang berbeda mengimplementasikan model data dengan menggunakan pemodelan khusus.

Hasil dari tahap ini berupa pernyataan DDK dalam bahasa DBMS terpilih yang merupakan skema level konseptual dan eksternal dalam sistem basis data. Tetapi jika pernyataan DDL termasuk beberapa parameter rancangan fisik, spesifikasi DDL yang lengkap harus menunggu setelah tahap rancangan basis data fisik selesai. Beberapa alat bantu CASE (computer-assisted software engineering) otomatis dapat membangkitkan DDb untuk sistem komersial dari rancangan skema konseptual.

2.3.5 Tahap 5 : Desain Basis Data Fisik

Perancangan basis data fisik adalah proses memilih struktur penyimpanan khusus dan mengakses path untuk file basis data untuk mendapatkan performansi yang baik pada aplikasi basis data. Setiap DBMS menawarkan berbagai pilihan organisasi file dan akses path. Termasuk di dalamnya berbagai tipe pengindeksan, clustering record yang berhubungan melalui pointer dan berbagai tipe hashing. Bila suatu DBMS dipilih, proses perancangan basis data fisik dibatasi pada struktur yang tepat untuk file basis data melalui pilihan yang ditawarkan DBMS. Kriteria berikut biasanya digunakan untuk menuntun pemilihan rancangan basis data fisik :

1. Waktu respon : Merupakan waktu antara pengiriman transaksi basis data untuk eksekusi dan penerimaan respon.
2. Utilitas ruang penyimpanan : Merupakan jumlah ruang penyimpanan yang digunakan file basis data dan struktur akses path pada disk, termasuk pengindeksan dan akses path lain.
3. Transaction throughput : Merupakan jumlah transaksi rata-rata yang dapat diproses per metnin, merupakan parameter kritis dari sistem transaksi seperti yang digunakan pada reservasi pesawat atau bank.

Hasil dari rancangan basis data fisik dalam tahap ini merupakan ketentuan awal dari struktur penyimpanan dan akses path untuk file basis data. Juga selalu diperlukan modifikasi rancangan berdasarkan observasi performansi setelah sistem basis data diimplementasikan. Aktifitas tahap berikutnya adalah tuning basis data.

2.3.6 Tahap 6 : Implementasi Basis Data dan Tuning

Setelah rancangan logika dan fisik selesai, kita dapat mengimplementasikan sistem basis data. Hal ini merupakan tanggung jawab DBA bersama desainer basis data. Pernyataan dalam DDL (data definition language) termasuk SDL (storage definition language) dari DBMS terpilih dikompilasi dan digunakan untuk membuat skema basis data dan file basis data (kosong). Basis data dapat kemudian dipopulasikan dengan data. Jika data diubah dari sistem komputerisasi sebelumnya, rutin konversi diperlukan untuk format kembali data untuk menyimpan ke basis data baru.

Transaksi basis data harus diimplementasikan dengan aplikasi yang dibuat programming berdasarkan spesifikasi konseptual dari transaksi dan kemudian menulis dan melakukan uji coba kode program dengan perintah DML. Jika transaksi siap dan data disimpan ke basis data, tahap rancangan dan implementasi selesai dan tahap operasi dari sistem basis data dimulai.

RINGKASAN:

- Terdapat aturan system informasi dalam organisasi, system basis data dilihat sebagai bagian system informasi dalam aplikasi berskala besar.
- Basis data sebagai manajemen resource informasi dalam organisasi dan kelangsungan hidupnya harus tetap berjalan.
- Terdapat 6 tahap dalam proses perancangan. Tiga tahap yang umum dalam rancangan basis data adalah rancangan konseptual, rancangan logika (pemetaan model data) dan rancangan fisik. Sedangkan tahap inisial adalah koleksi dan analisa kebutuhan yang biasanya termasuk dalam tahap pra desain.
- Ada beberapa criteria organisasi dalam pemilihan DBMS
- Jika permasalahan performansi terdeteksi dan aplikasi baru diaplikasikan, rancangan harus dimodifikasi
- Pada basis data relasional, factor yang berakibat pada keputusan rancangan basis data fisik dan menyediakan tuntunan pemilihan alternative rancangan desain

LATIHAN SOAL :

1. Sebutkan 6 tahap perancangan basis data!
2. Manakah dari 6 tahap tersebut sebagai aktifitas utama dalam proses perancangan basis data ? Mengapa ?
3. Mengapa perancangan skema dan aplikasi dilakukan secara parallel ?
4. Mengapa digunakan model data implementation-independent selama perancangan skema konseptual ?
5. Mengapa diperlukan koleksi dan analisa kebutuhan ?
6. Buatlah aplikasi actual dari suatu system basis data. Tentukan kebutuhan dari level pemakai yang berbeda dalam hal kebutuhan data, tipe query dan transaksi yang diproses.
7. Bagaimana karakteristik dari model data untuk rancangan skema konseptual harus diproses ?
8. Apa perbedaan dua pendekatan utama dalam rancangan skema konseptual
9. Strategi apa yang digunakan untuk merancang skema konseptual dari kebutuhan ?
10. Sebutkan langkah-langkah view integration ke rancangan skema konseptual.
11. Sebutkan factor untuk memperlancar pemilihan paket DBMS untuk system informasi dalam organisasi.
12. Apa yang dimaksud pemetaan data model system-independent ? Apa perbedaannya dengan system-dependent ?