



BAB 4 MANAJEMEN DATA

4.1. Sumber Data.

3 sumber data adalah dari: internal, eksternal, dan personal.

- **Internal.** Data disimpan dalam satu atau beberapa tempat dalam suatu organisasi. Data ini mengenai orang, produk, services, dan proses. Contoh: data mengenai karyawan dan penggajiannya.
- **Eksternal.** Mulai dari database komersial sampai data yang dikumpulkan dari sensor dan satelit. Bentuknya bisa berupa CD-ROM, film, musik, atau suara. Juga gambar, diagram, atlas, dan televisi. Pilihlah data eksternal yang relevan saja, karena banyak tak relevan dengan MSS yang diinginkan. Contoh: PDBI (dulu, jamannya Christianto Wibisono sebelum pindah ke Australia), berita/informasi dari BEJ/BES, Biro Pusat Statistik, dan lain-lain.
- **Personal.** Para pakar dapat memberikan kontribusinya pada MSS untuk pelbagai aplikasi tertentu. Contoh: perkiraan penjualan atau opini mengenai kompetitor.

4.2. Pengumpulan dan Permasalahan Data.

Metode pengumpulan data mentah bisa menggunakan cara:

- **Manual.** Contoh: metode time studies (selama observasi), survey (menggunakan kuisioner), observasi (misal dengan menggunakan kamera video), meminta pendapat pakar (misal dengan mewawancarainya).
- **Instrumen dan sensor.** Digunakan untuk membantu metode manual atau malah kadang dominan peranannya bila cara-cara manual sudah tak mampu lagi.

Permasalahan data mengikuti asas GIGO (Garbage In Garbage Out).

Problem	Typical Cause	Possible Solutions (in Some Cases)
Data are not correct.	Raw data were entered inaccurately. Data derived by an individual were generated carelessly.	Develop a systematic way to ensure the accuracy of raw data. Whenever derived data are submitted, carefully monitor both the data values and the manner in which the data were generated.
Data are not timely.	The method for generating the data is not rapid enough to meet the need for the data.	Modify the system for generating the data.
Data are not measured or indexed properly.	Raw data are gathered according to a logic or periodicity that is not consistent with the purposes of the analysis. A detailed model contains so many coefficients that it is difficult to develop and maintain.	Develop a system for rescaling or recombining the improperly indexed data. Develop simpler or more highly aggregated models.
Needed data simply do not exist.	No one ever stored data needed now. Required data never existed.	Whether or not it is useful now, store data for future use. This may be impractical because of the cost of storing and maintaining data. Furthermore, the data may not be found when they needed. Make an effort to generate the data or to estimate them if they concern the future.

4.3. Pelayanan Database Komersial.

Pelayanan database online (komersial) menjual akses ke database yang besar (biasanya meliputi negara). Ini dapat menambahkan data eksternal ke MSS untuk waktu tertentu pada biaya yang layak. Yang diperlukan adalah: terminal komputer, modem, telepon, password, dan biaya pelayanannya.

Pelayanan database online dikembangkan secara terpisah satu sama lain, dengan pelbagai perbedaan pada bahasa perintah yang dipakai, struktur file, dan protokol akses. Jika ditambahkan juga disini kompleksitas pencarian data, proliferasi dari database online (mungkin ribuan), dan kesulitan dalam hal standarisasi; maka jelas diperlukan knowledge lebih lanjut untuk menggunakan database ini lebih efisien. ES (biasanya dikombinasikan dengan NLP) dapat digunakan sebagai antarmuka dengan database seperti itu.



Contoh pelayanan database komersial yang cukup baik adalah:

- CompuServe and The Source.
- CompuStat.
- Data Resources, Inc.
- Dow Jones Information Service.
- Interactive Data Corporation.
- Lockheed Information Systems.
- Mead Data Central.

4.4. Database dan Manajemen Database.

DBMS didesain untuk sebagai suplemen yang memungkinkan kita dalam mengintegrasikan data dalam skala yang lebih besar, struktur file yang kompleks, pengambilan, perubahan dan penampilan data secara cepat, dan keamanan data yang lebih baik dari sekedar database biasa.

Software Database.

- **Procedural Languages.** Contoh: BASIC, COBOL, FORTRAN, dan Pascal. Diikuti dengan Object-oriented Language: C++, Delphi, Java, C#, PHP, dll.
- **Nonprocedural Fourth-generation Language (4GLs).** Bahasa utama dalam kebanyakan generator DSS dan pelbagai tool MSS lainnya.
- **Problem-oriented Language.** Bahasa ini memungkinkan programmer untuk menjelaskan karakteristik masalah yang akan diselesaikan bukannya prosedur-prosedur yang harus diikuti. Contoh: GPSS (General Purpose Simulation Software) digunakan untuk mengkonstruksi model simulasi dan dapat digunakan untuk mengkonstruksi DSS yang besar.

4GLs memiliki keuntungan:

- Result-oriented.
- Meningkatkan produktivitas paling sedikit 5 kali lipat sampai sekitar 300 kali lipat untuk pelbagai aktivitas.
- Sebagian besar end-user dapat membangun sistem dengan 4GLs tanpa bantuan perantara, sebab 4GLs dirancang baik untuk spesialis maupun end-user.

Fourth-generation languages (4GLs) digunakan untuk membangun sistem secara cepat dan murah. Sehingga merupakan tool pengembangan yang efektif.

- Sebagai tool yang digunakan saat suatu MSS atau MSS generator (engine) dibangun, dan dibangunnya itu adalah dari sekedar konsep kasar.
- Sebagai basis dalam pembangunan tool atau komponen yang lain dari suatu MSS; sebagai contoh, satu DBMS dapat dibuat dengan 4GL.
- Sebagai suatu MSS generator untuk membangun aplikasi tertentu.
- Sebagai tool ideal untuk pusat informasi (Information Centers).

4.5. Fourth-generation Systems.

DBMS yang digunakan sebagai tool pengembangan dari suatu DSS, biasanya ditulis dalam 4GL dan diintegrasikan dengan pelbagai elemen yang lain.

Contoh dari sistem seperti ini adalah suatu komposisi populer spesial untuk komputer mainframe dan disebut dengan Fourth-Generation System (FGS) yang lengkap. FGS yang lengkap mempunyai pelbagai fitur yang membuat user mudah berkomunikasi dengan komputer, dan buat pembangunan DSS membuatnya mudah membangun suatu DSS. Fitur-fitur ini adalah:

- Fourth-generation DBMS.
- Nonprocedural report writer (atau report generator).
- Nonprocedural language untuk data maintenance.
- Screen definition dan management facility.
- Graphic enhancement.
- Query language.
- Relational language.
- Applications management.
- Client/server management.
- Extended data access.
- Modeling language.
- Environment untuk applications development.
- Environment untuk information consumers.
- Micro-to-mainframe environment.



4.6. Struktur Database dan SQL.

Relasi diantara pelbagai record individu yang tersimpan dalam database dapat dinyatakan dengan pelbagai struktur logikal. DBMS didesain dengan menggunakan struktur ini untuk mengeksekusi fungsi-fungsinya.

Relational Database.

Beberapa file data "direlasikan" dengan field data dari dua (atau lebih) file data. Keuntungan dari bentuk ini adalah user mudah untuk mempelajari, data mudah dikembangkan atau diubah, dan mudah diakses dalam pelbagai format tanpa perlu mengantisipasinya pada waktu awal mendesain dan mengembangkannya.

Hierarchical.

Menyusun item data dalam gaya top-down, membuat link logikal diantara item data yang berelasi. Sehingga mirip seperti pohon, atau bagan organisasi.

Network.

Struktur ini mengijinkan link yang kompleks, termasuk koneksi rumit diantara item-item yang berelasi. Struktur ini disebut dengan model CODASYL. Ini dapat menghemat penyimpanan data dengan men-share-nya pada beberapa item.

SQL.

Bahasa data yang menjadi standar untuk mengakses dan memanipulasi data dalam RDBMS. Digunakan untuk akses online ke database, untuk pelbagai operasi DBMS dari program, dan untuk fungsi-fungsi administrasi database.

Juga digunakan untuk mengakses dan memanipulasi fungsi dari produk software DBMS terkemuka saat ini (contohnya: Oracle, DB2, Ingres, dan Supra).

SQL adalah bahasa nonprocedural dan user-friendly, sehingga end-user dapat menggunakannya dalam query dan pelbagai operasi database.

4.7. Object-oriented Database.

Aplikasi MSS yang kompleks, misalnya yang melibatkan CIM (Computer Integrated Manufacturing) membutuhkan akses ke data yang kompleks, yang melibatkan gambar-gambar dan relasi yang juga kompleks.

Arsitektur database hierarchical, network, ataupun relasional tak dapat mengatasi database yang kompleks tersebut secara efisien. Walaupun SQL digunakan untuk membuat dan mengakses database relasional, solusinya tetap saja tiak efektif. Semua arsitektur ini berhubungan dengan database alphanumeric (huruf, angka, dan karakter-karakter lain), padahal terkadang diperlukan representasi grafis untuk mendapatkan hasil terbaik.

Manajemen data berorientasi objek didasarkan pada prinsip-prinsip pemrograman berbasis objek. Sistem OODB mengkombinasikan karakteristik bahasa pemrograman berbasis objek seperti Smalltalk, C++, C#, Delphi, Java, PHP, dan lain-lain dengan mekanisme penyimpanan dan pengaksesan data. Tool-tool berorientasi objek berfokus secara langsung pada database. OODB mengijinkan kita untuk menganalisis data pada level konseptual yang menekankan hubungan alamiah diantara objek. Abstraction digunakan untuk membuat hirarki inheritance, serta object encapsulation mengijinkan desainer database untuk menyimpan baik data konvensional maupun kode-kode prosedural dalam objek yang sama. Contoh dari object-oriented data manager: GemStone, VBase, G-Base, Express, Ontos, dan Versant.

4.8. Aplikasi Database dan Spreadsheet.

Hal utama yang ada di DMBS adalah manajemen data, yaitu untuk membuat, menghapus, mengubah, dan menampilkan data. DBMS mengijinkan user untuk meng-query data seperti halnya menghasilkan report.

Program spreadsheet berhubungan dengan aspek pemodelan DSS. Ini membantu membuat dan mengatur model, menampilkan kalkulasi berulang pada variabel yang berhubungan, serta melibatkan faktor-faktor matematis, statistik dan keuangan.

Sayangnya ada kebingungan terhadap sifat utama DMBS dan program spreadsheet ini. Utamanya disebabkan karena pelbagai DMBS itu menawarkan kemampuan yang serupa dengan yang ada di spreadsheet terintegrasi, seperti: Lotus 1-2-3 dan Excel, sehingga membuat user dapat menampilkan kerja spreadsheet dengan suatu DBMS. Serupa dengan fenomena ini, banyak program spreadsheet menawarkan pelbagai kemampuan DBMS.

Aplikasi DSS membutuhkan data dan pemodelan dalam prosesnya, sehingga DSS dapat dibangun dengan mengembangkan lebih lanjut DBMS atau dengan mengintegrasikan spreadsheet. Alternatif lain, dapat pula dibangun dengan sebagian DBMS dan sebagian lagi dari spreadsheet. Pendekatan ketiga adalah dengan menggunakan DSS generator yang benar-benar terintegrasi.



Akhirnya dapat disimpulkan bahwa seseorang dapat membangun DSS dari konsep kasar dan tak perlu menggunakan 4GLs yang ada.

4.9. Enterprise DS dan Information Warehouse.

Terdapat dua kontradiksi dalam bisnis modern. Pertama, adanya kebutuhan terhadap solusi hardware dan software yang khusus dan terlokalisasi. Kedua, adanya kebutuhan untuk mengefektifkan cost dari penyatuan semua sumber informasi kedalam aset bisnis yang termanajemen.

Manajer harus mengantisipasi tantangan ini, mereka harus bisa mengantisipasi pertumbuhan dan makin kompleksnya keragaman peralatan dan sistem. Sistem enterprise jelas makin kompleks.

Sistem komputer yang melibatkan keseluruhan organisasi disebut dengan enterprise computing atau enterprise-wide systems. Istilah enterprise mengacu pada pengertian keseluruhan organisasi.

Enterprise computing adalah suatu arsitektur dari sistem komputer terintegrasi yang melayani bermacam-macam kebutuhan suatu enterprise. Ini adalah kerangka kerja berbasis teknologi yang terdiri dari pelbagai aplikasi, hardware, databases, jaringan, dan tool-tool manajemen, dan biasanya berasal dari pelbagai vendor.

Keuntungan dari enterprise computing adalah:

- Menyediakan pelayanan yang responsif dan handal secara kontinyu.
- Paduan kerjasama yang lebih baik dalam penggabungan solusi client/server baru pada mainframe yang sudah ada. Proses penyatuan ini dapat mencegah pelbagai aplikasi kritis dan mengembangkan misi yang sudah ada.
- Sering dan cepatnya perubahan dan bertambahnya kompleksitas dapat diakomodasikan dengan cepat, tanpa mengganggu ketersediaan sistem dan jaringan.
- Optimasi yang lebih tinggi pada resources jaringan dan sistem memastikan bahwa pelayanan kualitas yang tinggi dijaga pada biaya terendah yang mungkin.
- Otomatisasi proses manajemen menjadikan biaya administrasi dan operasional sistem berkurang seiring dengan pertumbuhan enterprise.
- Keamanan jaringan dan data meningkat.

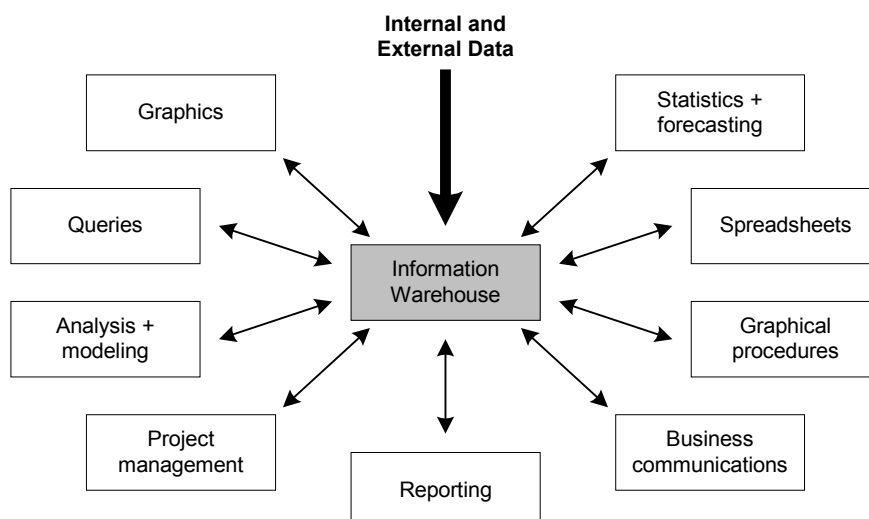
4.9.1. Konsep Information Warehouse (IW).

Kunci sukses dari enterprises-wide MSS adalah infrastruktur yang mendukung akses, retrieval, manipulasi, analisis, konstruksi presentasi, metodologi penampilan grafis, dan komunikasi (transfer) hasil-hasil serangkaian aktivitas yang dilakukan.

IBM membuat konsep IW yang didesain untuk meng"unlock" data secara ekonomis pada suatu enterprise, sampai saat dibutuhkan; yang selama ini belum bisa dilakukan oleh user MSS. IW adalah sekumpulan DBMS, interface, tool, dan pelbagai fasilitas yang mengatur dan menyampaikan informasi yang handal, tepat waktu, akurat, dan mudah dimengerti dalam pengambilan keputusan bisnis.

Di bawah ini disajikan bagan dari IW-nya IBM:

Decision Support Systems-Definition





4.10. Arsitektur Client/Server.

Arsitektur yang mencoba untuk mengorganisasi PC, LAN, dan mainframe, ke dalam sistem yang fleksibel, efektif, dan efisien. Dicitrakan oleh:

- Client-nya adalah PC atau workstation, dihubungkan ke jaringan, yang digunakan untuk mengakses resources jaringan.
- User dihubungkan oleh interface ke Client. Umumnya menggunakan GUI.
- Pada sistem client/server terdapat pelbagai client, masing-masing dengan interface user sendiri-sendiri. Client ini saling berbagi resources yang disediakan server.
- Server adalah mesin yang melayani client dengan pelbagai pelayanan seperti database, tempat penyimpanan yang besar, atau pelayanan komunikasi ke jaringan.
- Server bisa berupa workstation yang besar, sebuah mainframe, minikomputer, dan/atau peralatan LAN.
- Client dapat menampilkan pelbagai query, command, dalam bahasa yang sudah umum digunakan semisal SQL sebagai presentasi ke server.
- Client dapat mengirimkan query atau command ke server untuk pelbagai tugas yang tak dapat diproses di client (atau lebih cepat diproses di server).
- Server menampilkan hasil-hasilnya pada layarnya client.
- Kebanyakan server adalah: database server, file server, print server, image-processing server, computing server, dan communication server.
- Server tidak memulai suatu pekerjaan, ini hanyalah sekedar reaksi dari permintaan client.
- Server tak dapat berkomunikasi dengan setiap client untuk menentukan apa yang harus dilakukan untuk setiap permintaan client tertentu.
- Pada client/server computing, transaksi client (dieksekusi pada desktop computer) bekerjasama dengan service (dieksekusi pada komputer yang lebih besar).
- Taks/tugas dibagi atas 2 bagian: bagian front-end diselesaikan oleh client, dan bagian back-end diselesaikan oleh server. Client menampilkan manipulasi data lokal dan user interface-nya. Server menangani database dan pemrosesan transaksi yang lebih intensif.
- Server melayani file sharing, penyimpanan dan retrieval informasi, manajemen jaringan dan dokumen, dan fungsi-fungsi perantara (untuk aliran informasi internal dan eksternal), seperti misalnya manajemen e-mail, BBS, dan video text.

Aplikasi client/server dibagi menjadi 4 kategori:

- Aplikasi pesan, seperti e-mail.
- Penyebarluasan suatu database diantara pelbagai jaringan komputer.
- Sharing/pemakaian bersama pada file/peralatan lain, atau remote akses komputer.
- Pemrosesan aplikasi intensif dimana job dibagi dalam pelbagai task, masing-masingnya dikerjakan pada komputer yang berbeda.

Keuntungan dari arsitektur client/server dapat dilihat pada tabel di bawah ini:

Feature	Benefit
Networked webs of small, powerful machines	If one machine goes down, your business stays up
Computer arrays with thousands of MIPS; clients aggregate MIPS beyond calculation	The system provides the power to get things done without monopolizing resources. End-users are empowered to work locally
Some workstations are as powerful as mainframe, but cost 90% less	By giving you more power for less money, the system offers you the flexibility to make other purchases or to increase your profits
Open systems	You can pick and choose hardware, software, and services from various vendors
Systems grow easily and are infinitely expandable	It's easy to modernize your system as your needs change
Individual client operating environments	You can mix and match computer platforms to suit the needs of individual departments and users

4.11. Multidimensionality.

Ringkasan data dapat diorganisasi dalam pelbagai cara untuk kepentingan analisis dan presentasi. Pendekatan ini disebut dengan multidimensionality. Keuntungan dari pendekatan ini adalah data diorganisasi menurut cara pandang manajer bukan menurut cara pandang analisis sistem, juga pelbagai presentasi data yang sama dapat diatur dengan mudah dan cepat. 3 faktor dalam multidimensionality: dimensi, ukuran, dan waktu.



- Contoh dimensi: produk, salespeople (orang-orang bagian penjualan), segmen pasar, unit bisnis, lokasi geografis, jalur distribusi, negara, atau industri.
- Contoh ukuran: uang, sales volume (volume penjualan), keuntungan penyimpanan, aktual vs perkiraan.
- Contoh waktu: harian, mingguan, bulanan, caturwulan, atau tahunan.

Contohnya, seorang manajer ingin mengetahui penjualan dari produk M pada area geografis tertentu, yang dilakukan oleh orang di bagian penjualan tertentu, selama bulan tertentu juga, yang dihitung berdasarkan unit. Jika jawaban dari pertanyaan ini dapat disediakan tanpa memperhatikan struktur databasenya, maka hal itu dapat dilakukan lebih cepat, dan dapat dilakukan oleh user itu sendiri. Ini bisa terjadi jika data diorganisasikan dalam database multidimensional atau jika produk software tersebut didesain berdasar konsep multidimensionality.

Pendekatan multidimensionality ini biasanya lebih banyak digunakan dalam EIS. Tapi perlu diingat, pendekatan ini harus dibayar dengan hal-hal berikut:

- Database multidimensional memakan tempat 40 persen lebih banyak dibandingkan dengan database relasional yang sudah diringkaskan.
- Produk multidimensional membutuhkan biaya 50 persen lebih banyak dibandingkan dengan produk relasional standar.
- Waktu proses pemanggilan databasenya membutuhkan resources dan waktu, tergantung pada volume data dan jumlah dimensinya.
- Antarmuka dan perawatannya lebih kompleks dibandingkan dengan database relasional.

4.12. Pengaksesan Data: Data Dipping.

Selama beberapa waktu, teknologi informasi berkonsentrasi pada pembangunan sistem bermisi kritis – sistem yang mendukung pemrosesan transaksi perusahaan (Corporate Transaction Processing). Sistem ini membutuhkan toleransi kesalahan tertentu dan juga menyediakan respon sistem yang cepat. Solusi berikutnya adalah penyediaan sistem yang mendukung Online Transaction Processing (OLTP), yang bertumpu pada lingkungan database relasional terdistribusi. Pengembangan selanjutnya adalah penambahan pada arsitektur client/server. Contoh aplikasi OLTP ini: INFORMIX-OnLine.

Akses ke data sering dibutuhkan baik oleh aplikasi OLTP maupun oleh DSS dan juga oleh MSS lainnya. Sayangnya menyediakan layanan untuk kedua tipe ini menimbulkan pelbagai masalah. Sehingga beberapa perusahaan memilih untuk memisahkan databasenya ke dalam tipe OLTP dan DSS. Yang lain mencoba untuk menggunakan tool-tool yang inovatif untuk pengaksesan data; dimana tool ini berbasiskan PC yang murah, jaringan, GUI, dan pendekatan client/server. Beberapa orang menyebut hal ini data dipper, sebab sistem ini mengizinkan user untuk mengaduk-aduk apa saja di dalam database untuk mendapatkan apa yang mereka perlukan. Tool ini ditujukan untuk mendayagunakan user. Sehingga, aplikasi jenis ini lebih memperhatikan tentang bagaimana mendapatkan (retrieve) dan menampilkan data, daripada bagaimana data diakuisisi atau disimpan.

Data dipping ini memiliki juga nama lain: Business Intelligence Systems, end-user data access, dan data access and reporting tools. Contoh produknya:

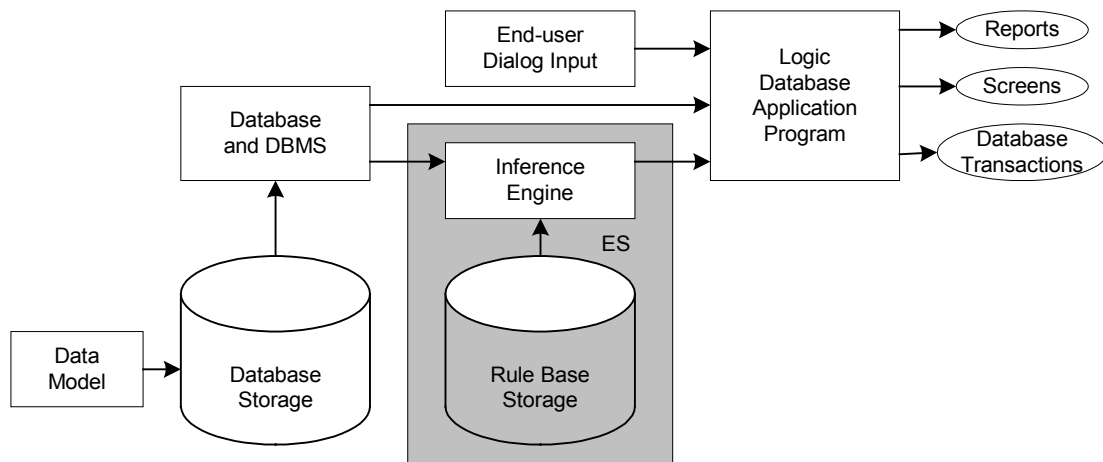
- Excel (dari Microsoft), diperkuat dengan Q&E (dari Q&E Corp.).
- Visual Basic (dari Microsoft).
- Lotus 1-2-3 (dari Lotus Development Corp., sekarang dibeli Microsoft), diperkuat dengan DataLens.
- LightShip (dari Pilot Software).
- Personal Access (dari Spinnaker Corp.).
- Quest (dari Gupta Corp.).
- Forest and Trees (dari Channel Computing, sekarang merupakan divisi dari Trinzic Corp.).

4.13. Intelligent Database.

Organisasi, pribadi dan publik, terus menerus mengumpulkan data, informasi, dan knowledge, dan menyimpannya ke dalam sistem terkomputerisasi. Updating, reviewing, penggunaan, dan penghapusan informasi ini makin lama makin kompleks seiring dengan bertambahnya jumlah data.

Pengembangan aplikasi MSS membutuhkan akses ke database. Sebagai contoh, tanpa akses database tentulah menjadi sulit untuk menggunakan ES dalam aplikasi MIS yang besar seperti otomasi pabrik dan otorisasi kartu kredit.

Teknologi AI, khususnya ES, dapat mempermudah manajemen database. Salah satu cara untuk melakukan itu adalah dengan memperkuat sistem manajemen database dengan penyediaan kemampuan inferencing. Pendekatan ini disebut dengan intelligent database. Di bawah ini terdapat diagramnya:



4.14. Kesimpulan.

- Data terdapat dalam sumber-sumber internal, eksternal, dan personal.
- Data eksternal tersedia dalam ribuan database komersial online, kamus (dictionaries), direktori, report, dan lain-lain.
- Data untuk MSS perlu dikumpulkan berulang-ulang di lapangan menggunakan salah satu dari pelbagai metode.
- Data untuk MSS kemungkinan mempunyai pelbagai masalah seperti: data yang tidak benar (incorrect data), data yang waktunya tidak tepat (nontimely data), data yang ukuran dan indeksinya tidak bagus (poorly measured and indexed data), terlalu banyak data, atau malah tak ada data sama sekali.
- Database online yang besar seperti CompuServe dan Dow Jones Information Service bisa menjadi sumber utama data MSS.
- DSS dapat diprogram menggunakan bahasa pemrograman generasi ketiga, tetapi biasanya diprogram dengan menggunakan bahasa pemrograman generasi keempat.
- Sistem generasi keempat mengikutsertakan pelbagai fitur terintegrasi untuk manajemen data.
- Data diorganisasi dalam pelbagai cara, baik itu berarsitektur relasional, hirarkikal, maupun jaringan. Untuk kebanyakan MSS, tipe yang disukai adalah relasional.
- SQL adalah akses standar untuk database relasional.
- Terdapat kecenderungan untuk menjadikan MSS terdistribusi melalui jaringan.
- MSS terdistribusi mengumpulkan kelebihan-kelebihan dari PC dan kekuatan dari suatu mainframe.
- Pelbagai MSS ditawarkan pada sistem client/server.
- OODB disediakan khusus untuk menangani DSS yang kompleks seperti yang terdapat pada Computer Integrated Manufacturing.
- OODB mudah untuk digunakan dan sangat cepat dalam aksesnya, sangat berguna dalam MSS terdistribusi.
- Banyak perusahaan mengembangkan pendekatan berskala global (enterprise-wide) dalam hal manajemen data. Sebagai contoh adalah Information Warehouse dari IBM.
- Enterprise-wide information system mengacu pada sistem yang menyediakan komunikasi diantara semua karyawan secara organisasional. Juga menyediakan akses ke sembarang data atau informasi yang diperlukan oleh semua karyawan pada pelbagai lokasi.
- Dalam client/server, beberapa PC (client) saling terhubung satu sama lain dan terhubung juga ke database, telekomunikasi, dan pelbagai penyedia layanan yang lain (server).
- Data multidimensionality mengijinkan kita untuk melihat (view) data dengan cepat dengan dimensi yang berbeda, walaupun data tersebut berada pada file dan database yang berbeda.
- Satu dari tujuan kritis yang utama adalah untuk membuat database intelligent, sehingga user dapat menemukan sendiri suatu informasi dengan cepat.