

PERKEMBANGAN DATABASE

Obyektif :

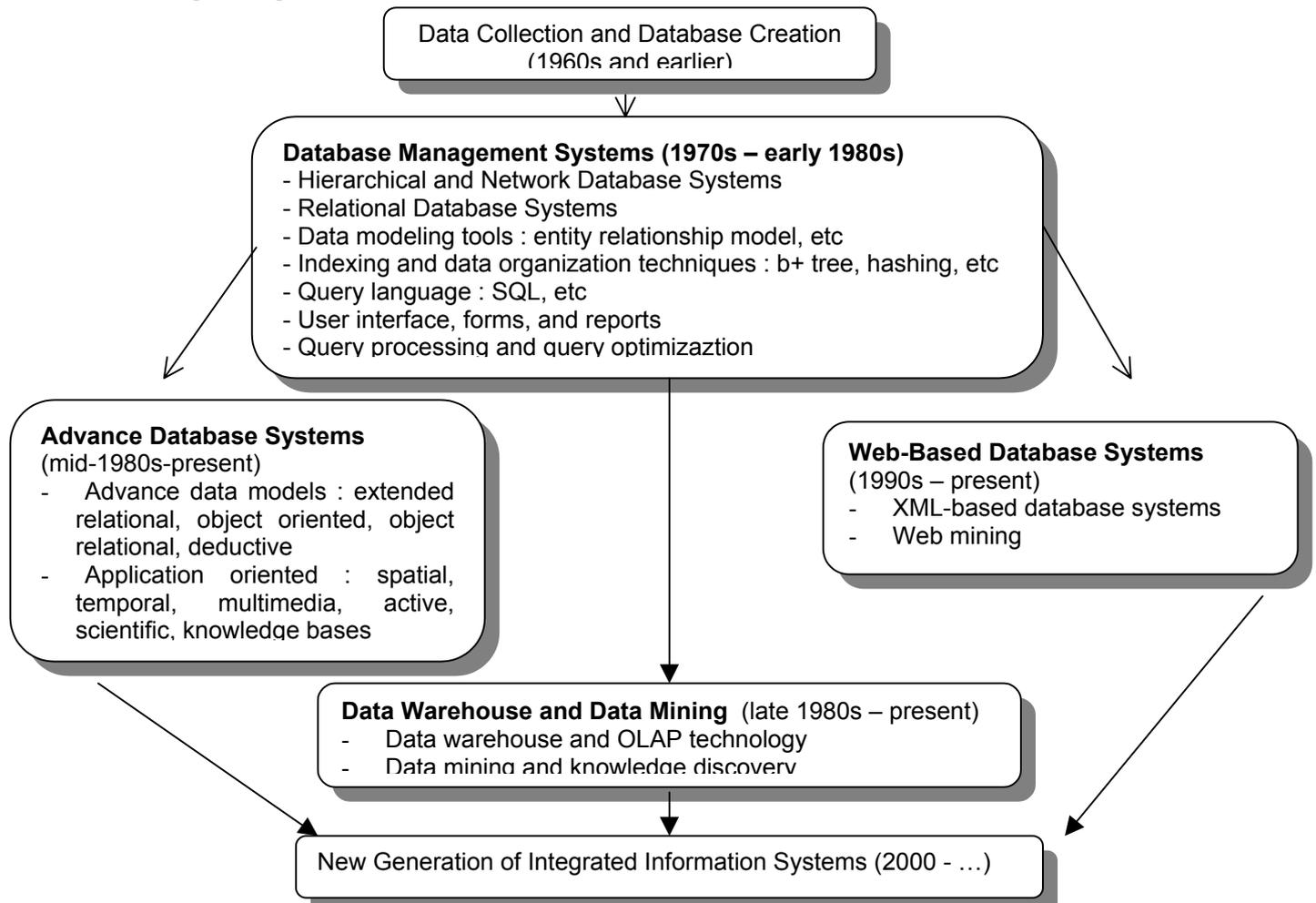
- Mahasiswa mengetahui perkembangan database yang ada saat ini.
- Mahasiswa mengetahui konsep Object-Relational Database, Data Warehouse, Data Mart, Data Mining, dan Web Database

PERKEMBANGAN BASIS DATA SAAT INI

Sejak tahun 1960-an penggunaan basis data sudah digunakan untuk bidang komersial, dimana pemrosesan file-nya masih berbasis manajemen file tradisional. Perkembangan komputer yang semakin pesat diikuti dengan perkembangan perangkat lunak untuk aplikasi bisnis, sejak tahun 1970-an sampai awal tahun 1980 manajemen berbasis file tradisional berkembang menjadi manajemen basis data.

Di dalam manajemen basis data dikenal berbagai model data yang dapat digunakan untuk mendeskripsikan sebuah data dalam merancang suatu basis data. Manajemen ini memungkinkan banyak *pemakai* untuk mengakses data secara bersamaan sehingga fasilitas yang dimiliki oleh manajemen sudah semakin banyak yaitu fasilitas pemanipulasian data, kontrol konkurensi data, recovery data, keamanan data dan didukung dengan fasilitas komunikasi data karena manajemen ini sudah terhubung dengan suatu jaringan .

Perkembangan dunia usaha semakin meningkat ditunjang dengan perkembangan komunikasi yang mempermudah organisasi atau perusahaan untuk mengakses data, sehingga mengubah manajemen basis data menjadi manajemen basis data tingkat lanjut didukung dengan fasilitas data warehousing dan fasilitas basis data berbasis web sebagai salah satu strategi organisasi dalam meningkatkan kinerja dan keuntungan organisasi.



Gambar 1. Evolution of Database Technology

A. ORDBMS

A.1. ORDBMS (*Object Relational Database Management System*)

Penggunaan teknologi RDBMS pada awalnya sangat dominan, tetapi dikarenakan fasilitas-fasilitas yang ada pada RDBMS tidak lagi sesuai dengan aplikasi tingkat lanjut, maka RDBMS digantikan dengan OODBMS. Pada OODBMS terdapat banyak sekali bentuk-bentuk object oriented seperti metode *encapsulation*, *inheritance*, *polymorphism*, dll.

Model data relasional lanjutan tidak hanya ada satu, tetapi terdapat berbagai macam model data, dimana karakteristik dan tingkat kebutuhan data yang telah dibuat. Bagaimanapun semua model data yang akan digunakan mengacu pada konsep objek dan mempunyai kemampuan untuk menyimpan data di dalam database.

Berbagai macam terminologi yang digunakan untuk sistem model relasional tingkat lanjut yaitu ERDBMS. Tapi sekarang ini untuk sistem berbasis objek dapat digunakan OODBMS.

Inti dari kinerja RDBMS yaitu ada pada optimasi query-nya dan juga pengetahuan mengenai bagaimana mengeksekusi fungsi dari *user-define* secara efisien, mengambil keuntungan dari pengindeksan pada struktur yang baru, memetakan query dengan cara baru, dan menavigasi antara data menggunakan referensi data.

Penggunaan OODBMS untuk suatu organisasi yang sangat besar dan universal tidak lagi sesuai sehingga untuk mendukung kinerja dari organisasi tersebut dibutuhkan suatu ORDBMS (*Object Relational Database Management System*).

ORDBMS memiliki fasilitas untuk mendefinisikan data yang kompleks, menspesialisasikan struktur indeks dibutuhkan untuk mengefisienkan pengambilan data. ORDBMS digunakan untuk dua sampai tiga dimensional data.

A.2. Perbandingan antara OODBMS dan ORDBMS

Pemodelan Data		
Model	OODBMS	ORDBMS
Identitas Objek	ada	Ada dengan tipe REF (References)
Pengkapsulan	Ada, tapi tidak dapat digunakan untuk query	Ada dengan UDT (User Defined Types)
Penurunan	Ada	Ada (dipisahkan antara UDT dan Table)
Polymorphism	Ada, sebagai object oriented pada model bahasa pemrograman	Ada
Objek kompleks	Ada	Ada, dengan UDT
Relasi	Ada	Sangat mendukung untuk mendefinisikan batasan referensial integritas

B. Data Warehousing

B.1. Konsep Data Warehousing

Konsep dasar dari data warehousing adalah informasi yang dikumpulkan dalam suatu gudang penyimpanan dan merepresentasikan solusi untuk pengaksesan data didalam sistem non relasional. Sehingga data warehousing dapat disebut sebagai database yang berorientasi pada subyek, terintegrasi, mempunyai *Time Variant* dan bersifat *non volatile* untuk kumpulan data yang mendukung dalam pengambilan keputusan.

- Berorientasi subyek artinya mengatur semua subyek utama pada suatu organisasi yang memfokuskan pada basis datanya bukan pada aplikasi yang digunakan untuk pengambilan keputusan.
- Terintegrasi artinya pemakaian data bersama-sama sering menyebabkan data tidak konsisten sehingga cara pandang *pemakai* terhadap data menjadi tidak sama. Agar terintegrasi pembentukan sumber data harus standar dan konsisten.
- *Time Variant*, data yang ada pada gudang hanya valid dan akurat pada titik waktu tertentu atau interval tertentu.
- *Non-volatile*, data tidak di-*update* secara *real time* tetapi selalu diperbaharui dari sistem operasi pada database yang ada.

B.2. Keuntungan dari Data Warehousing

1. Hasil yang diperoleh dari investasi lebih tinggi
2. Kompetitif
3. Meningkatkan produktivitas perusahaan

Jenis database yang tersimpan di dalam media penyimpanan data berdasarkan penggunaan data :

- Database yang memiliki data sering di-update disebut data OLTP (Online Transaction Processing). Data OLTP sering juga disebut data operasional, mencerminkan sifat aplikasi database yang dinamik.
- Database yang memiliki data sering digunakan untuk query disebut DSS (Decision Support System). Data DSS sering disebut data analitikal, mencerminkan sifat aplikasi database yang relatif statik.

Data Operasional	Data DSS
Berorientasi pada aplikasi : data digunakan untuk proses bisnis. Sebagai contoh : sistem perbankan dengan file terpisah yang sudah dalam bentuk normal untuk setiap proses bisnis.	Berorientasi pada subyek : data digunakan untuk subyek bisnis, misal informasi nasabah. Data dalam bentuk denormalisasi dimana sebuah record dapat meliputi keseluruhan proses bisnis.
Data terperinci	Data ringkas
Struktur statik	Struktur dinamik
Target operator komputer	Target pengambil keputusan pada seluruh tingkatan
Volatile (data dapat diubah)	Non volatile (data tidak bisa diubah setelah dimasukkan)
Kebutuhan data selalu diketahui sebelum rancangan sistem	Kebutuhan data sama sekali tidak diketahui sebelum rancangan sistem
Mengikuti siklus hidup pengembangan klasik dimana iterasi rancangan diselesaikan melalui normalisasi data, dan memeriksa kebutuhan pemakai	Siklus hidup pengembangan sama sekali berbeda, dimana pemakai menggunakan aplikasi struktur data yang ada dan membuat rancangan siap untuk dianalisis
Performansi penting karena jumlah pemakai konkuren sangat besar dalam mengakses data	Masalah performansi lebih longgar Karena jumlah pemakai jauh lebih sedikit dalam mengakses data sehingga tidak ada masalah konkuren yang perlu diperhatikan.
Penggerak-transaksi (Transaction-driven)	Penggerak-analisis (Analysis-driven)
Data harus selalu tersedia untuk pemakai akhir (back up dan recovery harus terencana dengan baik)	Tidak mempunyai tingkat kebutuhan ketersediaan data yang sama (perencanaan back up dan recovery lebih longgar)
Mencerminkan situasi mutakhir	Mencerminkan nilai historis

C. Data Mart

Untuk mencapai suatu *data warehouse* kelihatannya merupakan suatu tantangan besar dan memang demikian. Bahkan begitu besarnya sehingga beberapa pakar menyarankan pendekatan yang lebih sederhana yaitu menerapkan sesuatu yang dinamakan *data mart*. *Data mart* adalah database yang berisikan data yang menjelaskan satu segmen operasi perusahaan. Misalnya perusahaan mungkin memiliki *data mart* pemasaran, *data mart* sumber daya manusia, dsb.

D. Data Mining

Istilah yang sering digunakan bersama-sama dengan *data warehouse* dan *data mart* adalah *data mining*. *Data mining* adalah proses menemukan hubungan dalam data yang tidak diketahui oleh pemakai. *Data mining* membantu pemakai dengan mengungkapkan berbagai hubungan dan menyajikannya dengan suatu cara yang dapat dimengerti sehingga dapat menjadi dasar pengambilan keputusan. *Data mining* memungkinkan pemakai “menemukan pengetahuan” pada database yang dalam sepengetahuannya tidak ada.

Contoh Data Mining :

Sebuah bank telah memutuskan untuk menawarkan reksadana kepada para pelanggannya. Manajemen bank ingin mengarahkan materi promosi pada segmen pelanggan yang memberikan potensi bisnis terbesar.

- **Data Mining Berdasarkan Verifikasi.** Pendekatan yang dilakukan oleh para manajer adalah mengidentifikasi karakteristik yang mereka yakin dimiliki oleh pasar sasaran. Misalkan para manajer ingin mengarah pada pasangan muda, berpenghasilan ganda, dan kaya. Query dapat dimasukkan ke dalam DBMS, dan catatan yang tepat dapat dipanggil. Pendekatan yang seperti itu, yang mulai dengan hipotesis pemakai tentang bagaimana data tersebut terstruktur, disebut **data mining berdasarkan verifikasi** (*verification-driven data mining*). Kekurangan pendekatan ini adalah proses pemanggilan kembali diarahkan sepenuhnya oleh pemakai. Informasi yang dipilih tidak lebih baik daripada pandangan pemakai terhadap data tersebut. Ini merupakan cara tradisional untuk bertanya pada database.
- **Data Mining Berdasarkan Penemuan.** Pendekatan lain memungkinkan sistem *data mining* mengidentifikasi pelanggan terbaik untuk promosi tersebut. Sistem itu menganalisis *database*, mencari kelompok-kelompok dengan karakteristik umum. Dalam contoh bank, sistem *data mining* mungkin mengidentifikasi bukan hanya kelompok pasangan muda lulusan universitas tetapi juga pasangan yang sudah pensiun yang bergantung pada jaminan sosial dan pensiun. Sistem *data mining* dapat melaksanakan analisis selangkah lebih jauh, dengan merekomendasikan satu set promosi yang diarahkan pada kedua kelompok tersebut.
- **Kombinasi Data Mining Verifikasi dan Penemuan.** Perkembangan *data mining* di masa depan akan mengkombinasikan pendekatan hipotesis dan penemuan. Perkembangan ini menggunakan penalaran yang sama yang mendasari konsep Sistem Pendukung Keputusan (*Decision Support System* – DSS). Konsep tersebut memungkinkan pemakai dan komputer bekerja sama untuk memecahkan suatu masalah. Pemakai menerapkan keahliannya dalam hal masalah, dan komputer melakukan analisis data yang canggih untuk memilih data yang tepat dan menempatkannya dalam format yang tepat untuk pengambilan keputusan.

E. Web Database

Pada pokok bahasan ini akan dijelaskan mengenai web sebagai aplikasi dari suatu basis data dan arsitektur yang akan digunakan, pengaksesan database dan keuntungan juga kelemahan dari web database ini, sebelum membahas lebih dalam tentang sebuah web database terlebih dahulu harus diketahui apakah yang dimaksud dengan internet, world wide web dan apa saja yang ada di dalamnya.

Internet adalah kumpulan dari interkoneksi jaringan komputer di seluruh dunia, yang dibuat secara terpisah, yang menghubungkan situs–situs komersial, pendidikan dan organisasi pemerintahan juga Internet Services Provider. Kemunculan dari internet ini memungkinkan lalu lintas informasi yang sangat cepat, dengan memberikan konektifitas, akses informasi dan pelayanan online untuk pemakai di seluruh dunia.

Internet dan World Wide Web (WWW) saling bergantung satu sama lain, dimana WWW merupakan sistem informasi terdistribusi yang berbasis *hypertext* yang memberikan kemudahan dengan hanya melakukan klik maka informasi akan muncul, dengan kata lain mengambil suatu informasi di internet dengan menggunakan hyperlink. Web terdiri dari jaringan komputer yang dapat bekerja sebagai *server* yang memberikan informasi, dan sebagai *client* yang biasanya disebut dengan browser yang meminta suatu informasi. Web browser merupakan program khusus untuk menterjemahkan instruksi-instruksi format terhadap teks dan gambar yang hasilnya kemudian disajikan kepada para pemakai. Web browser yang paling banyak digunakan pada saat ini, yaitu Netscape Navigator dan Microsoft Internet Explorer, dan Web Server yang paling banyak digunakan yaitu Apache, Netscape Communication Server dan Microsoft Internet Information Server..

Informasi pada Web disimpan di dalam sebuah dokumen. Dokumen–dokumen ini yang menjadi unsur utama di dalam Web dapat dinyatakan dalam beberapa tipe. Tipe yang paling populer adalah dokumen *hypertext* yang disusun menurut bahasa khusus seperti *Hypertext Mark Up Language* (HTML). Dokumen HTML ini terdiri atas teks, spesifikasi jenis huruf dan instruksi format lainnya, dan didalamnya juga terdapat *link* ke dokumen lainnya atau ke halaman lain pada dokumen yang sama dengan menggunakan suatu protokol yang disebut dengan *Hypertext Transfer Protocol* (HTTP). Dokumen dan lokasi antar dokumen ini diidentifikasi oleh sebuah alamat yang disebut *Uniform Resourcer Locator* (URL) .

WEB Sebagai Bentuk Aplikasi Basis Data

A. Kebutuhan akan Web-Integrasi DBMS

Ketika banyak vendor DBMS berkerja memberikan solusi untuk konektifitas basis data berbasis web, sebagian besar organisasi membutuhkan solusi yang lebih umum untuk mencegah agar mereka tidak terpaku pada satu teknologi saja.

Beberapa kebutuhan di bawah ini dapat menjadi acuan bagi suatu organisasi yang ingin menggunakan aplikasi basis data berbasis web yang lebih fleksibel yaitu :

1. Kemampuan mengakses data organisasi dengan aturan yang aman.

2. Data dan konektivitas vendor yang independen yang memperbolehkan kebebasan memilih DBMS saat ini dan masa yang akan datang.
3. Kemampuan antar muka basis data dengan web browser dan web server
4. Solusi konektivitas yang memberikan keuntungan untuk semua bentuk DBMS di semua organisasi.
5. Pendekatan open-arsitektur agar dapat digunakan dengan berbagai macam sistem dan teknologi.
6. Solusi untuk biaya konektivitas dengan memberikan skalabilitas pengembangan dan perubahan strategi, dan juga menolong pengurangan biaya pengembangan dan pemeliharaan aplikasi.
7. Mendukung transaksi yang diminta oleh banyak HTTP
8. Mendukung untuk sesi dan aplikasi berdasarkan otentikasi.
9. Menerima kinerja
10. Meminimalkan kelebihan administrasi
11. Mempunyai peralatan produktivitas untuk dapat dikembangkan, diperlihara dengan mudah dan cepat.

B. Arsitektur Web-DBMS

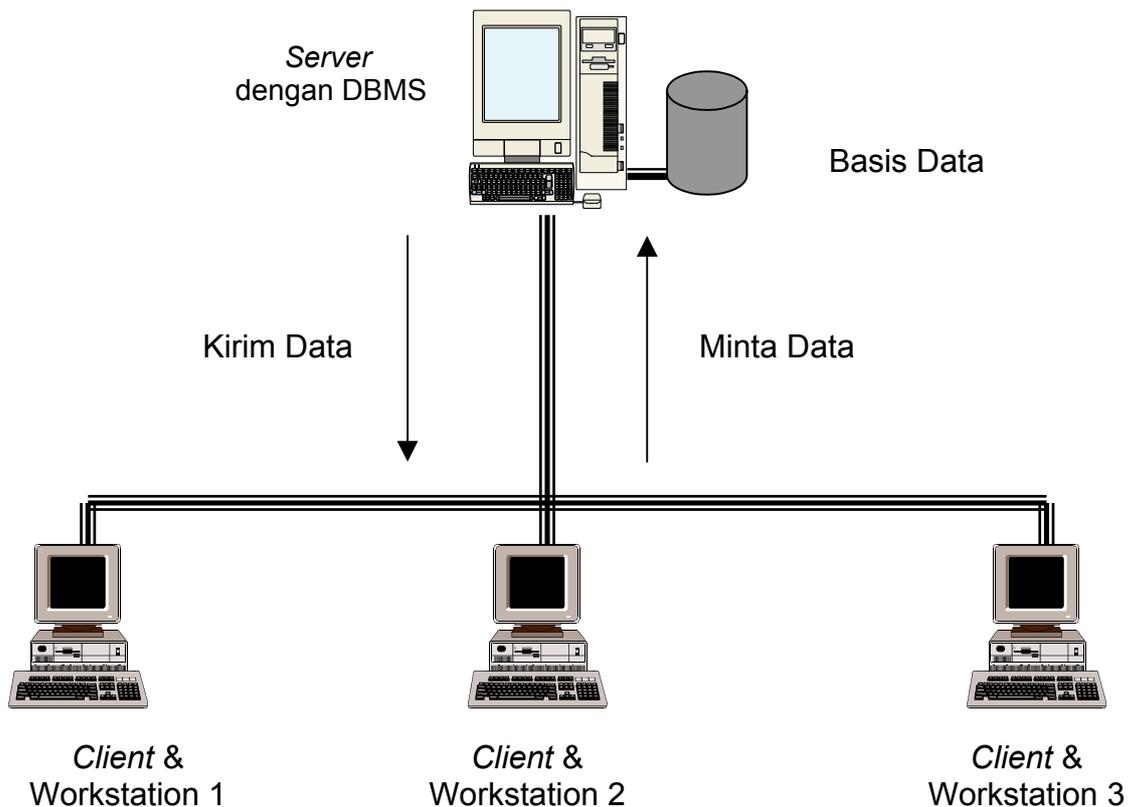
Aplikasi bisnis yang membutuhkan data yang intensif dapat terdiri dari empat komponen utama : basis data, transaksi logik, aplikasi logik dan antar muka untuk *pemakai*. Dalam lingkungan mainframe, komponen ini ditempatkan di satu tempat, seperti dapat ditemui pada lingkungan bisnis yang tersentralisasi. Untuk mengakomodasi pertumbuhan lingkungan bisnis yang desentralisasi, sistem *client-server* di kembangkan.

Ada dua macam implementasi sistem *client server* yaitu sistem *client server* sederhana (*2 tier*) dan sistem *client server* kompleks (*3 tier*).

1. Sistem *client server* sederhana (*2 tier*)

Sistem ini terdiri dari dua komponen utama yaitu *client* dan *server*. *Client* berisi aplikasi basis data untuk dapat memproses permintaan data pemakai sedangkan *server* berisi DBMS dan basis data yang dibutuhkan oleh *client* yang membutuhkan data. Setiap aktifitas yang dikehendaki oleh *pemakai* lebih dahulu ditangani oleh *client*. *Client* mengupayakan proses dapat ditangani sendiri tetapi jika melibatkan data yang tersimpan pada basis data maka *client* dapat melakukan kontak dengan *server*.

Sistem *client server* diterapkan pada sebuah jaringan komputer LAN namun jika terjadi peningkatan kompleksitas di dalam suatu organisasi yang mengakibatkan perubahan, perbaikan dan perkembangan dari aplikasi basis data maka akan menimbulkan masalah pada aktifitas pemasangan aplikasi yang tidak praktis karena harus melakukan instalasi pada semua mesin *client* yang digunakan.



Gambar 2. Konfigurasi 2 tier

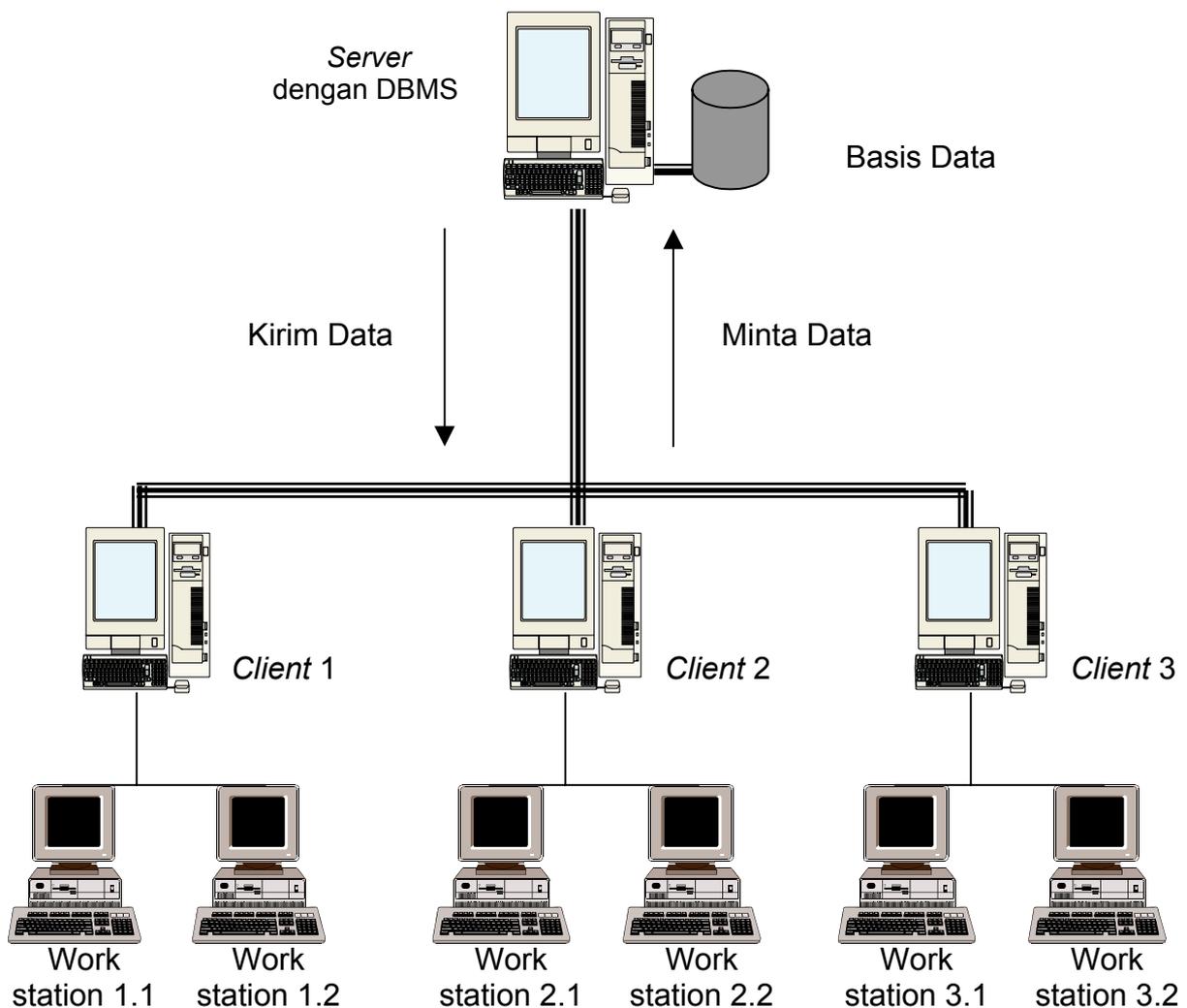
2. Sistem *client server* kompleks (3 tier)

Untuk menangani hal tersebut di atas maka dibangun arsitektur 3-tier untuk suatu organisasi yang lebih kompleks dengan menerapkan sistem jaringan yang lebar yaitu WAN. Pada arsitektur ini aplikasi basis data tidak ditempatkan di setiap workstation tetapi di pasang di setiap *client* yang jumlahnya lebih sedikit, dimana *client-client* ini ditempatkan pada lokasi yang lebih dekat dengan kumpulan workstation. Jadi setiap *client* dan workstation membentuk sebuah LAN tersendiri. Karena *client-client* ini merupakan basis untuk aplikasi basis data maka *client* dapat disebut dengan *Server* aplikasi.

Beberapa keuntungan jenis arsitektur ini adalah :

- Memungkinkan akses database yang besar
- Menaikkan performa
Jika *client* dan *server* diletakkan pada komputer yang berbeda, kemudian CPU yang berbeda dapat memproses aplikasi secara paralel. Hal ini mempermudah merubah mesin *server* jika hanya memproses database.
- Biaya untuk *hardware* dapat dikurangi
Hanya *server* yang membutuhkan *storage* dan kekuatan proses yang cukup untuk menyimpan dan mengatur database.

- Biaya komunikasi berkurang
Aplikasi menyelesaikan bagian operasi pada *client* dan mengirimkan hanya bagian yang dibutuhkan untuk akses database melewati jaringan, menghasilkan data yang sedikit yang akan dikirim melewati jaringan.
- Meningkatkan kekonsistenan
Server dapat menangani pemeriksaan *integrity* sehingga batasan perlu didefinisikan dan validasi hanya di satu tempat, aplikasi program mengerjakan pemeriksaan sendiri
- Map ke arsitektur *open-system* dengan sangat alami



Gambar 3. Konfigurasi 3 tier

C. Antar muka Web dengan basis data

Adanya antar muka antara web dan basis data menjadi penting dikarenakan dua alasan yaitu :

1. Bersamaan dengan adanya pertumbuhan perdagangan elektronik atau E-Commerce, pengolahan transaksi harus dapat dihubungkan dengan web. Para *pemakai* berinteraksi dengan basis data secara tidak langsung melalui serangkaian interaksi dokumen HTML yang kemudian dikirim sebagai reaksi untuk *server* dan selanjutnya digunakan sebagai parameter oleh program di *server* untuk melakukan pengolahan transaksi tertentu pada basis data.
2. Keterbatasan yang dimiliki oleh dokumen HTML, seperti :
 - a. Pemakaian dokumen web yang tetap tidak memungkinkan penampilan disesuaikan dengan keinginan *pemakai*, misalnya sebuah perusahaan mungkin menginginkan materi promosinya disesuaikan dengan keinginan pembeli.
 - b. Pada saat data di suatu basis data mengalami perubahan, maka dokumen–dokumen web yang melibatkan data yang sama menjadi tidak diperbaharui kembali jika tidak segera disesuaikan. Hal ini akan semakin merepotkan jika dokumen tersebut tersebar di sejumlah lokasi.

Untuk mengatasi keterbatasan ini maka dibuatlah suatu dokumen web yang lebih dinamis dengan terlebih dahulu membuat antar muka web dengan basis data. Sehingga pada saat dokumen diminta oleh *pemakai*, sebuah program di web *server* akan dieksekusi untuk menjalankan operasi queri ke basis data dan membuat dokumen sesuai dengan hasil queri tersebut. Dengan mekanisme ini maka informasi yang diinginkan oleh *pemakai* dapat terefleksi dengan baik.

Menyediakan Akses ke Basis Data pada WWW

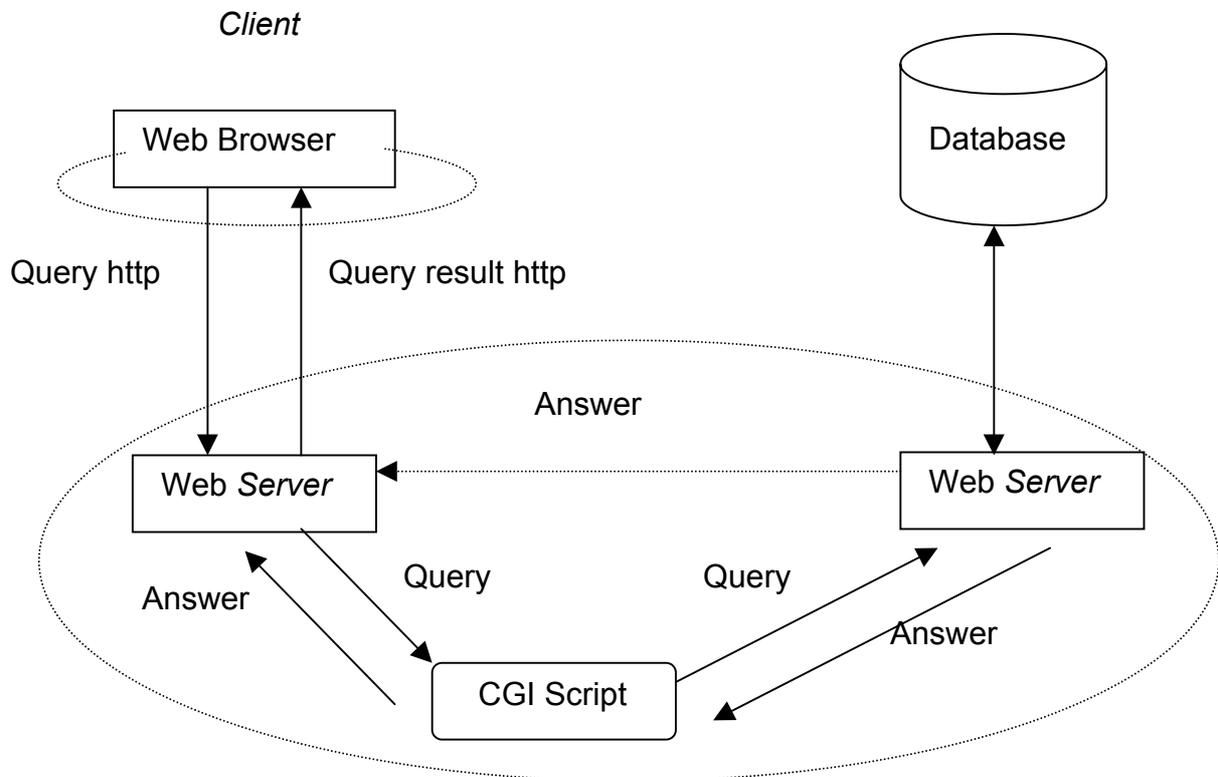
Teknologi pada saat ini bergerak dengan sangat cepat dan mempengaruhi pertumbuhan di segala bidang sehingga kebutuhan akan data pun semakin meningkat. Hal ini terlihat dengan munculnya perdagangan elektronik yang melakukan promosi dan transaksi di dunia maya, sehingga merubah proses operasionalnya dari statis ke halaman web yang lebih dinamis dengan arsitektur DBMS yang juga mengalami perubahan.

Umumnya Web *server* menggunakan *interface* standard yaitu *Common Gateway Interface* (CGI) sebagai *software* penengah antara Web *server* sebagai *front-end interface* dari pemakai dengan DBMS sebagai *back-end* nya, untuk memfasilitasi pengaksesan basis data yang heterogen/kompleks. CGI mengeksekusi program eksternal atau sebuah skrip untuk menghasilkan informasi yang dinamis dan dikirim ke *Server* dalam bentuk HTML untuk diberikan ke *web browser*. Selain CGI dapat digunakan JDBC sebagai *interface* yang digunakan untuk mengakses suatu basis data .

Dua pendekatan yang digunakan untuk mengakses sebuah basis data yaitu :

1. Akses dengan CGI Script ; menggunakan bahasa PERL, Tcl atau C
Penggunaan CGI kurang efisien karena harus melalui beberapa titik barulah dapat menghasilkan data yang diinginkan.

2. Akses dengan JDBC : menggunakan bahasa pemrograman Java, dimana JDBC ini sebagai penghubung *Object Conectivity Database* (ODBC) yang akan membentuk *Relational Database Management System* (RDBMS).



Keuntungan dan Kelemahan Web-DBMS

Web sebagai bentuk dari suatu sistem basis data dapat memberikan solusi yang inovatif untuk lingkungan inter- atau intra- pada jalannya bisnis di suatu organisasi, tetapi ada juga kelemahan yang dimiliki oleh Web-DBMS ini.

Keuntungan :

1. Kesederhanaan / Simplicity
2. Platform yang Independen
3. Graphical User Interface (GUI)
4. Standarisasi
5. Mendukung pertukaran antar platform
6. Akses yang transparan
7. Pemasangan instalasi yang diperhitungkan
8. Inovasi

Kelemahan :

1. Keandalan data
2. Keamanan data
3. Biaya
4. Skalabilitas
5. Keterbatasan fungsional dari HTML

6. Kestatisan
7. Kinerja
8. Peralatan untuk pembangunan yang tidak baik