

IMPLEMENTASI SISTEM BASIS DATA

- Pemilihan software (aplikasi, DBMS)
- Pemilihan hardware (server, client, infrastruktur jaringan)
- Pemilihan arsitektur jaringan
- Pemilihan brainware

Aplikasi Basis Data (1)

- Bagi pemakai mahir yang berinteraksi langsung terhadap basis data melalui DBMS, operasi basis data itu dapat berbentuk:
 - Penambahan data.
 - Pencarian data.
 - Pengubahan data.
 - Penghapusan data.
 - Pengurutan data.
 - Penggabungan data.
 - Penyimpulan/pengelompokkan data.
 - dan lain-lain.

Aplikasi Basis Data (2)

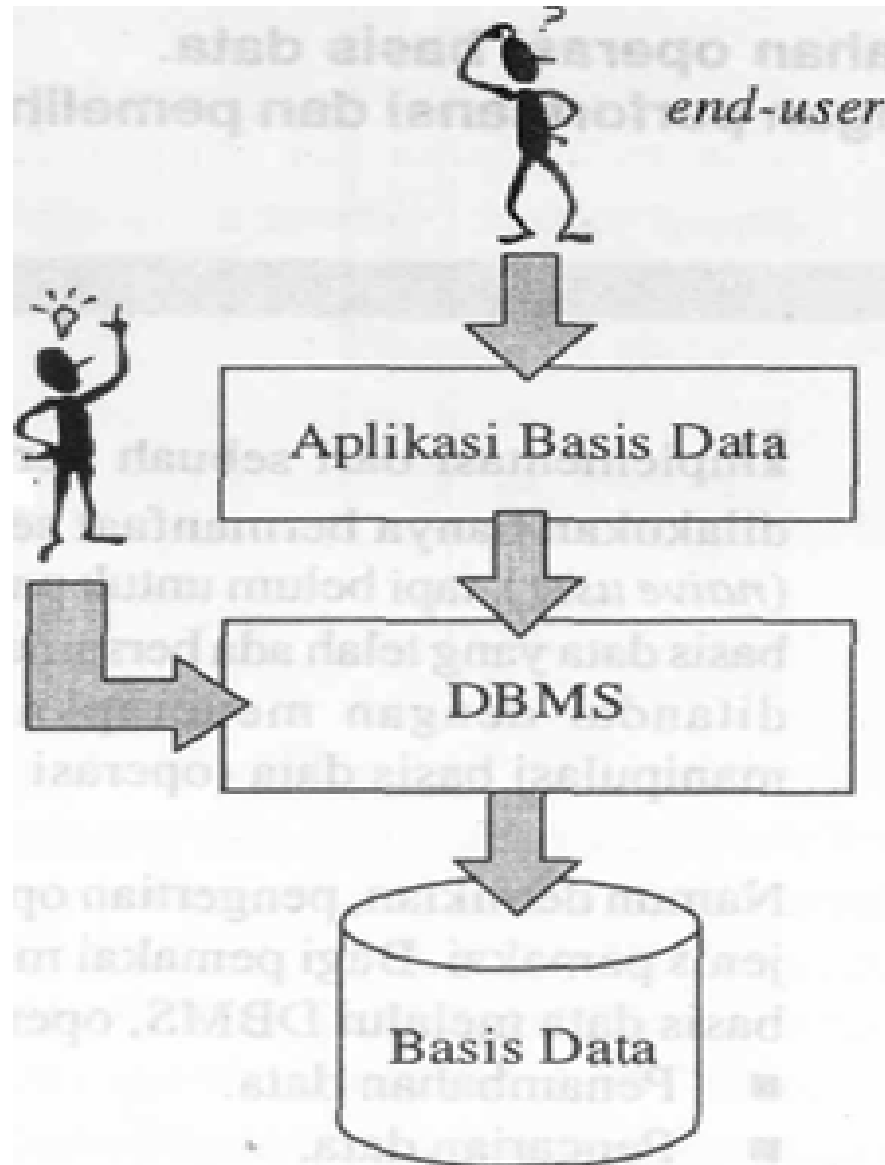
- Bagi pemakai akhir (*end-user*) yang memang tidak bersentuhan langsung dengan objek basis data, tetapi lebih banyak bersinggungan dengan aktivitas nyata yang memang terjadi di mana sistem basis data itu diterapkan, maka akan lebih mengenal operasi basis data seperti:
 - Pemasukan data (master barang, transaksi penjualan, jurnal-akuntansi, nasabah baru, dan lain-lain).
 - Monitoring data transaksi.
 - Pencetakan laporan harian/bulanan.
 - Penutupan data transaksi bulanan.
 - dan lain-lain.

Aplikasi Basis Data (3)

- Pada level implementasi, perbedaan tersebut dijumpai oleh adanya perangkat lunak (aplikasi) yang khusus dibuat untuk dapat digunakan oleh para pemakai akhir (*end-user*). Aplikasi ini akan menyediakan sejumlah operasi (menu) yang sesuai dengan berbagai aktivitas nyata yang memang dilakukan oleh para pemakai akhir (*end-user*).
- Selanjutnya, pilihan operasi *user* ini akan 'diterjemahkan' oleh aplikasi tersebut menjadi sejumlah operasi basis data elementer yang memang dapat dikenali/dikerjakan oleh DBMS. Aplikasi semacam itu kita sebut Aplikasi Basis Data.

Aplikasi Basis Data (4)

naive-
user



DBMS: Terpisah atau Menyatu dengan Aplikasi?

(1)

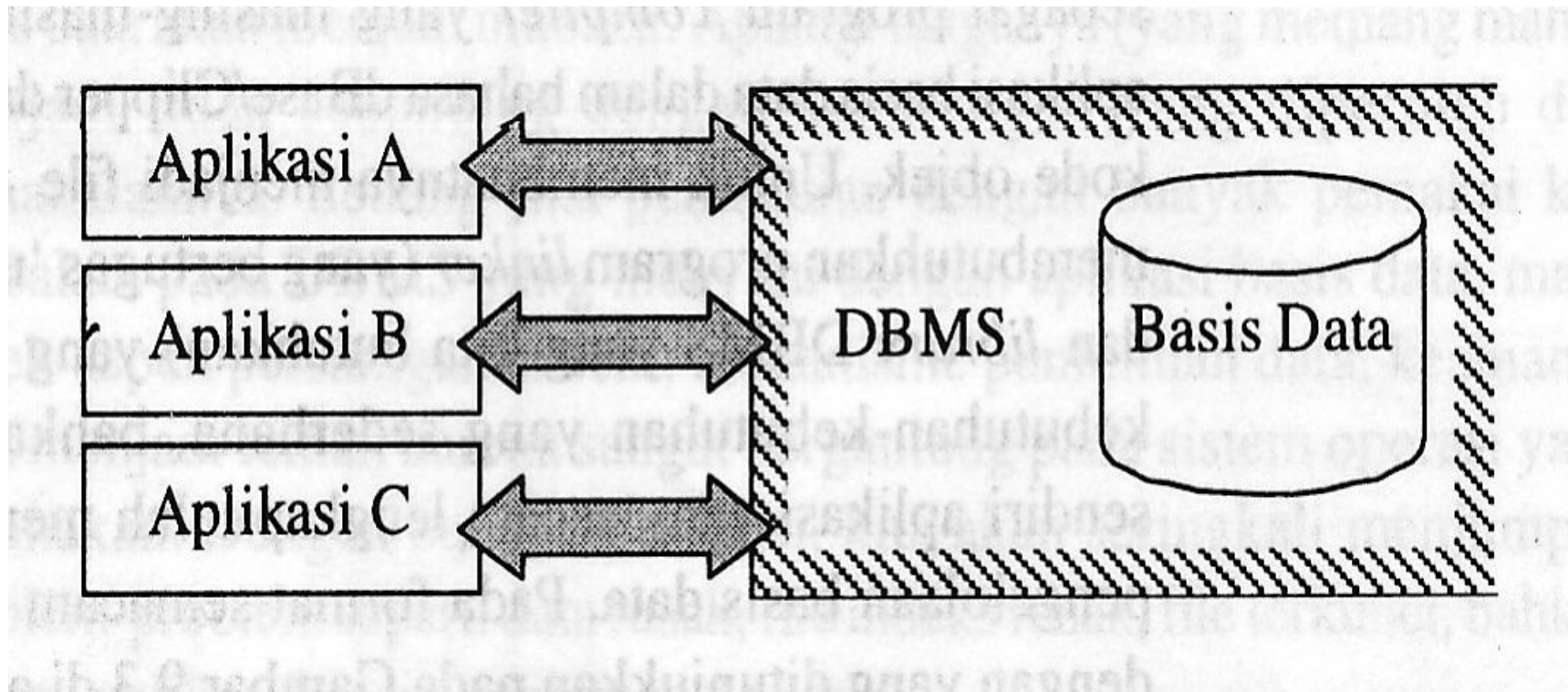
- Software untuk merancang aplikasi basis data: Java, VB.Net, ASP.Net, Delphi, MS.Access, Visual Foxpro, dll.
- DBMS yang besar (seperti Oracle, CA-OpenIngres, Sybase, Informix, IBM-DB2, SQL Server) memang dirancang sejak awal untuk berdiri sendiri dan terpisah dari aplikasi basis datanya.

DBMS: Terpisah atau Menyatu dengan Aplikasi?

(2)

- Karena berdiri sendiri, maka banyak sekali fasilitas (*feature*) yang ditangani oleh DBMS ini, seperti :
 - masalah pemeliharaan integritas basis data,
 - mekanisme *recovery* (pemulihan data otomatis akibat adanya kerusakan data),
 - mekanisme *backup periodik*,
 - pengendalian persaingan pemakaian data oleh banyak pemakai (*concurency control*),
 - mekanisme pengamanan pemakaian (*security mechanism*), dan lain-lain.

Gambar aplikasi basis data yang terpisah dari DBMS



DBMS: Terpisah atau Menyatu dengan Aplikasi?

(3)

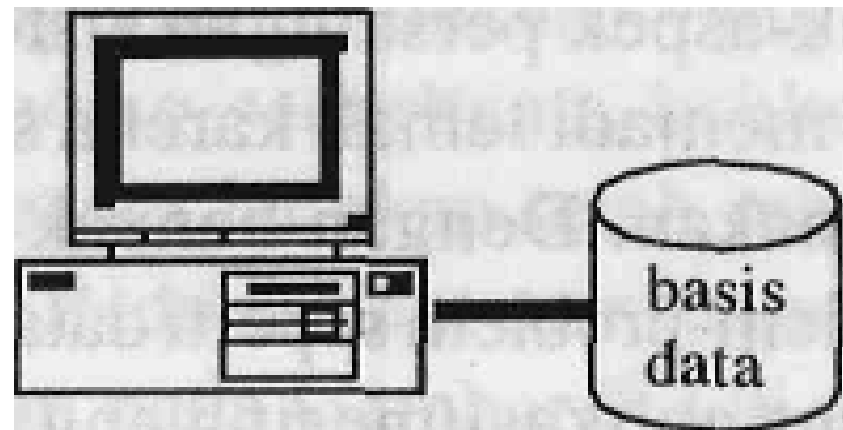
- Pada bentuk di atas, basis data dapat dianggap sebagai objek yang hidup, karena sesungguhnya aplikasi tidak pernah berinteraksi langsung dengan basis data. Tetapi selalu melalui DBMS sebagai perantara. Bahkan DBMS sendiri bisa melakukan aktifitas sendiri yang bisa ditangkap oleh aplikasi.
- Jadi interaksi antara aplikasi basis data dan DBMS merupakan interaksi dua arah (digambarkan dengan adanya tanda panah bolak-balik) antara dua perangkat lunak.

Arsitektur Sistem

- Arsitektur Sistem merujuk pada konfigurasi sistem secara keseluruhan yang akan menjadi 'tempat hidup' dari DBMS, basis data dan aplikasi yang memanfaatkannya.
- Bagaimana wujud dari 'tempat hidup' tersebut yang juga akan menentukan bagaimana para pemakai berinteraksi dengannya, sudah seharusnya ditetapkan sejak awal sebelum memulai perancangan basis data, atau paling tidak sebelum melaksanakan tahap implementasi basis data.
- Beberapa jenis arsitektur sistem yang dapat digunakan dan akan diuraikan pada bab ini adalah :
 - Sistem Tunggal/Mandiri (*Stand-Alone*)
 - Sistem Tersentralisasi (*Centralized System*).
 - Sistem Client-Server

Sistem Tunggal/Mandiri (*Stand-Alone*)

- Pada arsitektur ini, DBMS, basis data dan aplikasi basis data ditempatkan pada mesin (komputer) yang sama. Dengan demikian, pemakai yang dapat menggunakannya di setiap saat juga hanya satu orang (*single user*).
- Arsitektur ini merupakan arsitektur sistem yang paling sederhana dan paling murah. Arsitektur semacam ini dapat kita pilih dan gunakan, jika basis data yang dikelola memang tidak terlalu besar dan lebih bersifat membantu mempercepat pekerjaan-pekerjaan administratif.



Sistem Tersentralisasi (*Centralized System*) -1

- Kita tidak dapat mempertahankan pemakaian Sistem Mandiri, jika harus mengakomodasi pemakaian oleh banyak pemakai (*multi-user*).
- Arsitektur ini terdiri atas sebuah mesin *server* dan sejumlah terminal (yang menjadi tempat *user* berinteraksi dengan sistem). Yang tersentralisasi dalam arsitektur ini dapat mencakup basis data, DBMS dan aplikasi basis data atau basis data saja.

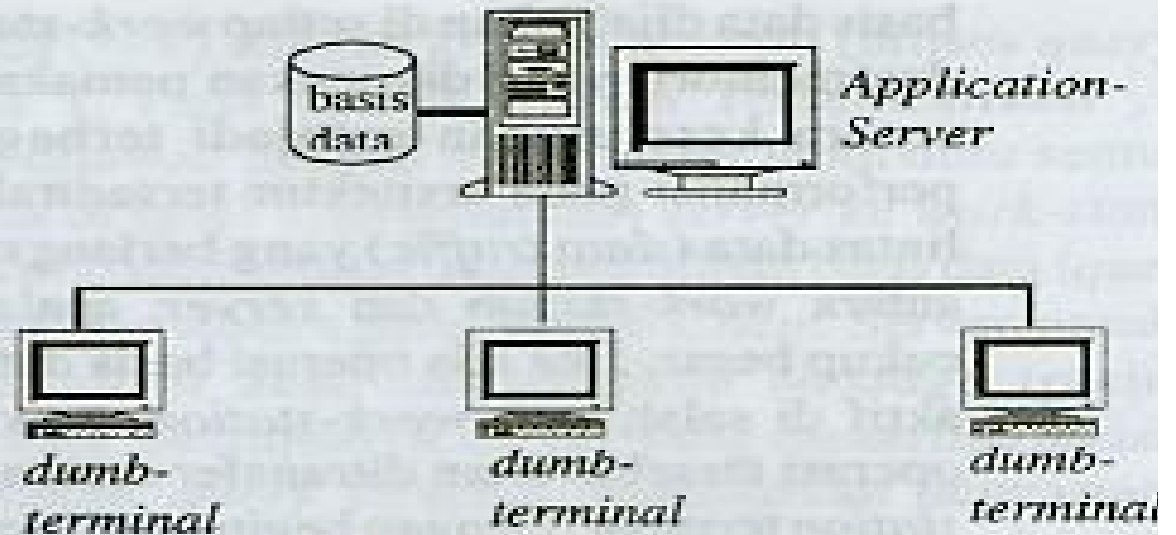
Sistem Tersentralisasi (*Centralized System*) - 2

- Untuk lingkup sentralisasi yang pertama, maka jenis *server-nya*. sering disebut sebagai *DBMS-server* atau *application-server* (server aplikasi) dan terminalnya lebih tepat disebut *dumb-terminal* (terminal pasif).
- Sedang jika yang disentralisasi hanya basis data, *server* yang kita gunakan biasa disebut *file server* dan terminalnya disebut *work-station* (stasiun kerja).

Sistem Tersentralisasi (*Centralized System*)-3

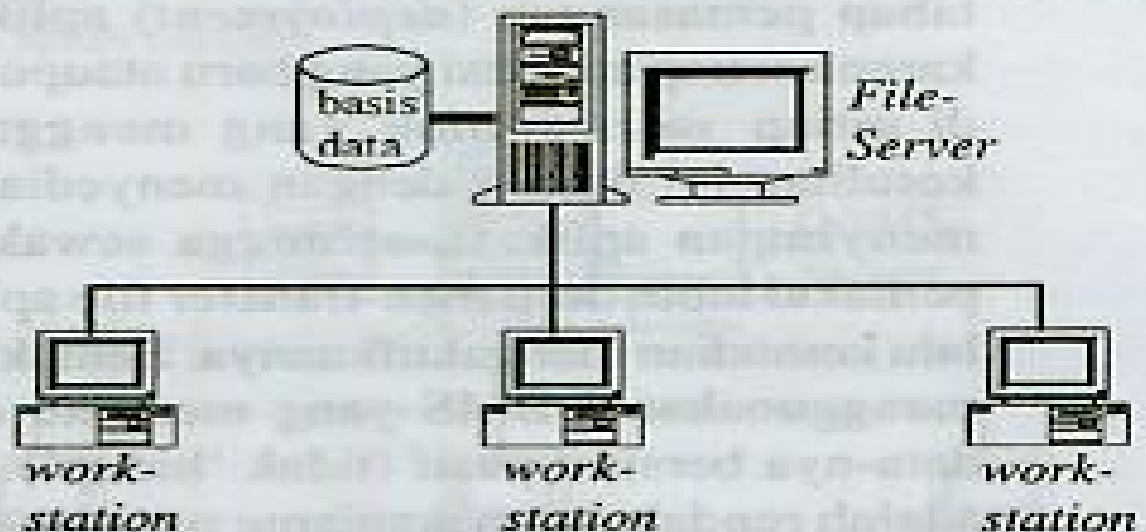
Gambar 9.6.a

Sentralisasi pada
Aplikasi dan Basis
Data



Gambar 9.6.b

Sentralisasi pada
Basis Data



Sistem Tersentralisasi (*Centralized System*)-4

- Pada bentuk sentralisasi yang pertama, beban *server* tentu saja sangat berat, karena digunakan secara bersama-sama oleh banyak pemakai untuk menjalankan aplikasi basis data dan DBMS (di samping tentu saja sistem operasi untuk jaringan).
- Untuk memperingan beban *server* (khususnya dalam alokasi sumber daya *server* seperti prosesor dan memori), bentuk sentralisasi ini umumnya menggunakan aplikasi berbasis teks (*text-base application*) dengan fasilitas antar-muka layar yang sederhana, sehingga ukuran aplikasi bisa lebih kecil.

Sistem Tersentralisasi (*Centralized System*)-5

- Pada bentuk sentralisasi yang kedua, beban *server* jauh lebih ringan, karena yang 'hidup' di situ hanyalah sistem operasi jaringan (*network operating system/NOS*) dan basis data dianggap setara dengan sebuah file atau beberapa file pada umumnya.
- Proses-proses aplikasi basis data dijalankan di setiap *work-station* (yang juga memiliki prosesor dan memori) yang digunakan pemakai. Karena itu, beban kerja sistem secara keseluruhan menjadi terbagi.

Sistem Client-Server -1

- Sebagaimana Sistem Tersentralisasi, arsitektur ketiga ini juga diterapkan pada sebuah sistem jaringan. Sistem Client-Server ini ditujukan untuk mengatasi kelemahan-kelemahan yang terdapat pada Sistem Tersentralisasi sebelumnya (baik sentralisasi pada aplikasi dan basis data maupun sentralisasi hanya pada basis data).

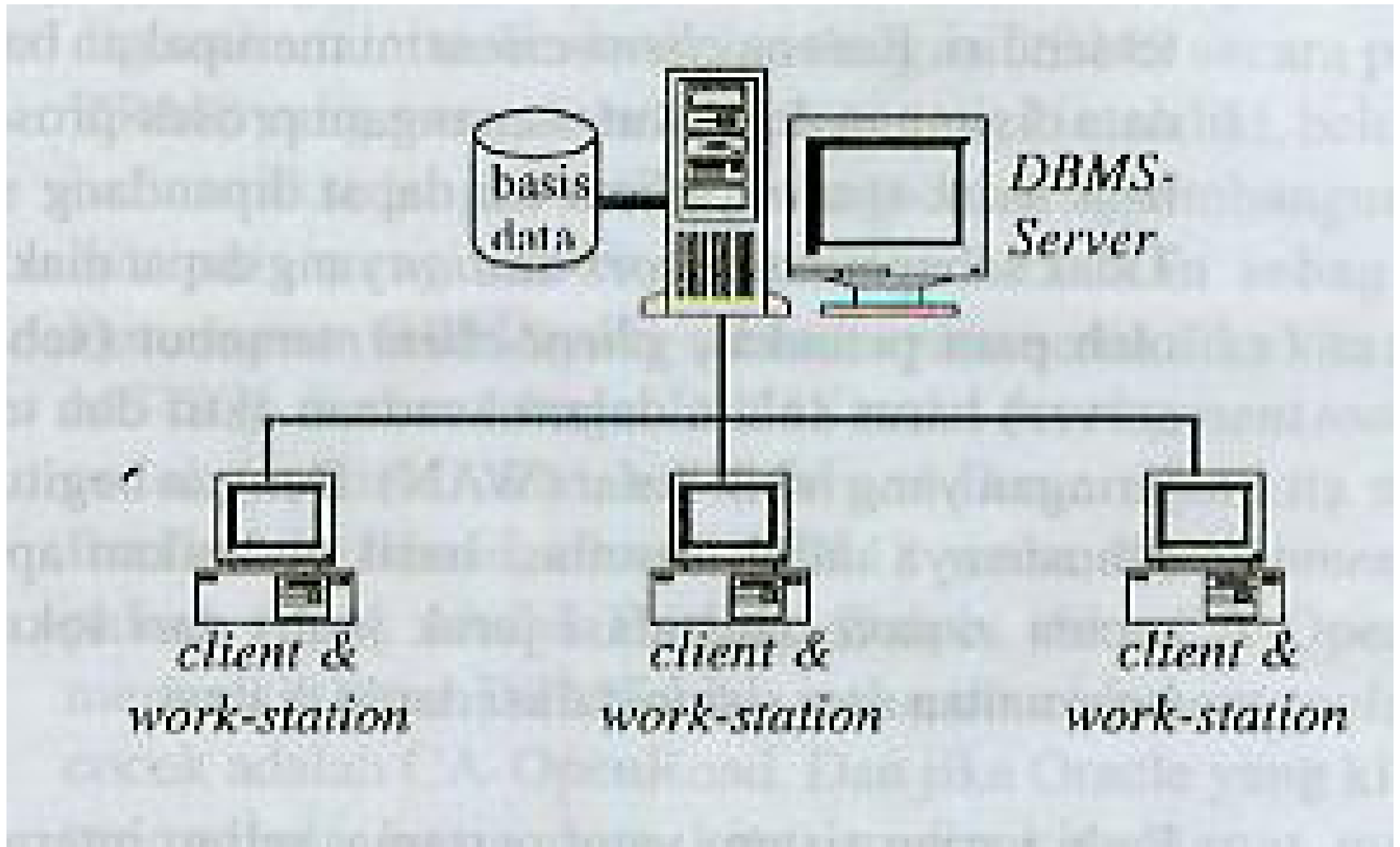
Sistem Client-Server -2

- Kelemahan pada bentuk Sistem Tersentralisasi yang pertama, yaitu beratnya beban *server* yang harus menangani semua proses, diatasi dengan membagi beban itu menjadi 2 bagian: *client* (yang menjalankan aplikasi basis data) dan *server* (yang menjalankan DBMS dan berisi basis data) pada mesin yang berbeda.

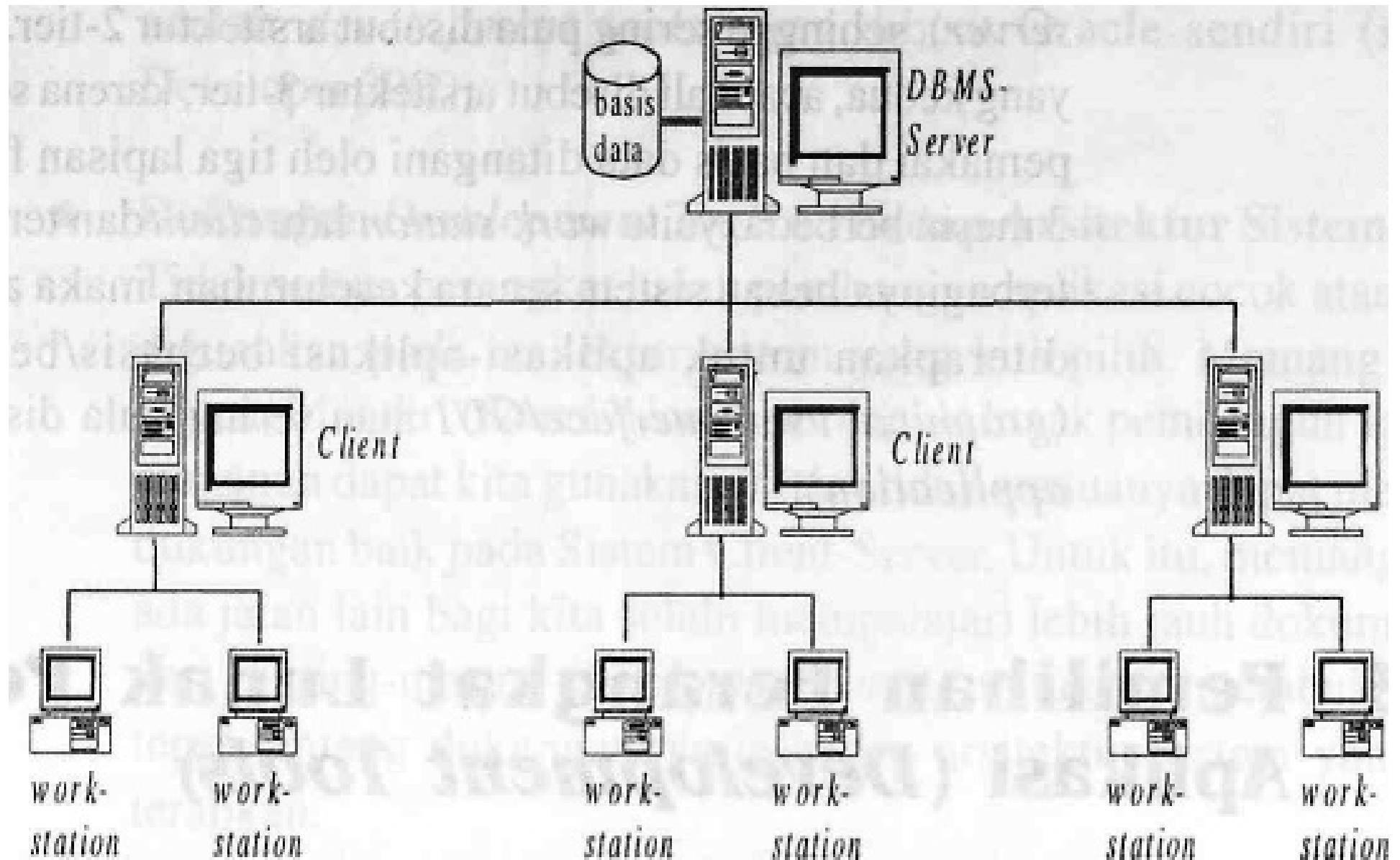
Sistem Client-Server -3

- Sedang kelemahan pada bentuk Sistem Tersentralisasi yang kedua, yaitu padatnya lalu lintas data antara *server* dan *work-station* diatasi dengan mekanisme transfer data yang lebih efisien.
- Sistem ini terdiri atas dua komponen (mesin) utama, yaitu *client* dan *server*. *Client* berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu ditangani oleh *client*. *Client* selanjutnya mengupayakan agar semua proses 'sebisa mungkin' ditangani sendiri. Jika ada proses yang harus melibatkan data yang tersimpan pada basis data, barulah *client* melakukan 'kontak' dengan *server*.

Gambar Sistem Client-Server sederhana



Gambar sistem Client-Server kompleks



Sistem Client-Server -4

- Dari kedua gambar di atas, dapat kita lihat adanya dua macam implementasi Sistem Client-Server. Bentuk yang pertama yang lebih sederhana dapat diterapkan pada sebuah jaringan komputer lokal (LAN), di mana fungsi *client* (untuk menangani sebagian besar proses pengolahan data seperti perhitungan, perulangan, perbandingan, dan lain-lain) dan fungsi *work-station* (untuk menangani interaksi dengan pemakai, menerima data masukan dan menayangkan hasil pengolahan) disatukan.