

Sistem Manajemen Basis Data

A series of horizontal lines in teal and light blue colors, with some lines having a 3D effect, extending from the left side of the slide towards the right.

Oleh : Rahmi Hidayati, S.Kom., M.Cs

Sistem Manajemen Basis Data

- Sistem manajemen basis data (*Database Management System-DBMS*) adalah perangkat lunak yang memudahkan organisasi untuk memusatkan data, mengolah data secara efisien dan menyediakan akses data bagi program aplikasi.
- DBMS bertindak sebagai antarmuka antara program aplikasi dan *file* data fisik.

Keunggulan DBMS

- Mengendalikan atau mengurangi duplikasi data.
- Menjaga konsistensi dan integritas data.
- Memudahkan perolehan informasi.
- Memaksakan penerapan standar.
- Menghemat biaya pemeliharaan program.
- Menanggulangi konflik kebutuhan antar pemakai.
- Meningkatkan tingkat respon dan kemudahan akses bagi pemakai akhir.
- Meningkatkan produktifitas pemrogram.

Kekurangan DBMS

- Kompleksitas yang tinggi membuat administrator dan pemakai akhir harus benar-benar memahami fungsi-fungsi dalam DBMS agar dapat memperoleh manfaat yang optimal.
- Ukuran penyimpanan yang dibutuhkan oleh DBMS sangat besar dan memerlukan memori yang besar agar bisa bekerja secara efisien.
- Harga DBMS yang handal sangat mahal.
- Terkadang DBMS meminta kebutuhan perangkat keras dengan spesifikasi tertentu.

Komponen Lingkungan *Database*

- Perangkat keras.
- Perangkat lunak.
- Data.
- Prosedur.
- Orang.

Orang-orang yang terlibat dalam DBMS

- Orang yang berperan langsung.
 - **Database Administrator** : orang yang bertanggung jawab terhadap administrasi penggunaan sumber daya *database*.
Tugasnya :
 - Mengatur otorisasi akses terhadap basis data
 - Memonitor penggunaan basis data
 - **Database desainer** : orang yang bertanggung jawab dalam perancangan basis data.
Tugasnya :
 - Mengidentifikasi data yang akan disimpan dalam basis data
 - Memilih struktur yang sesuai dalam menyajikan dan menyimpan data

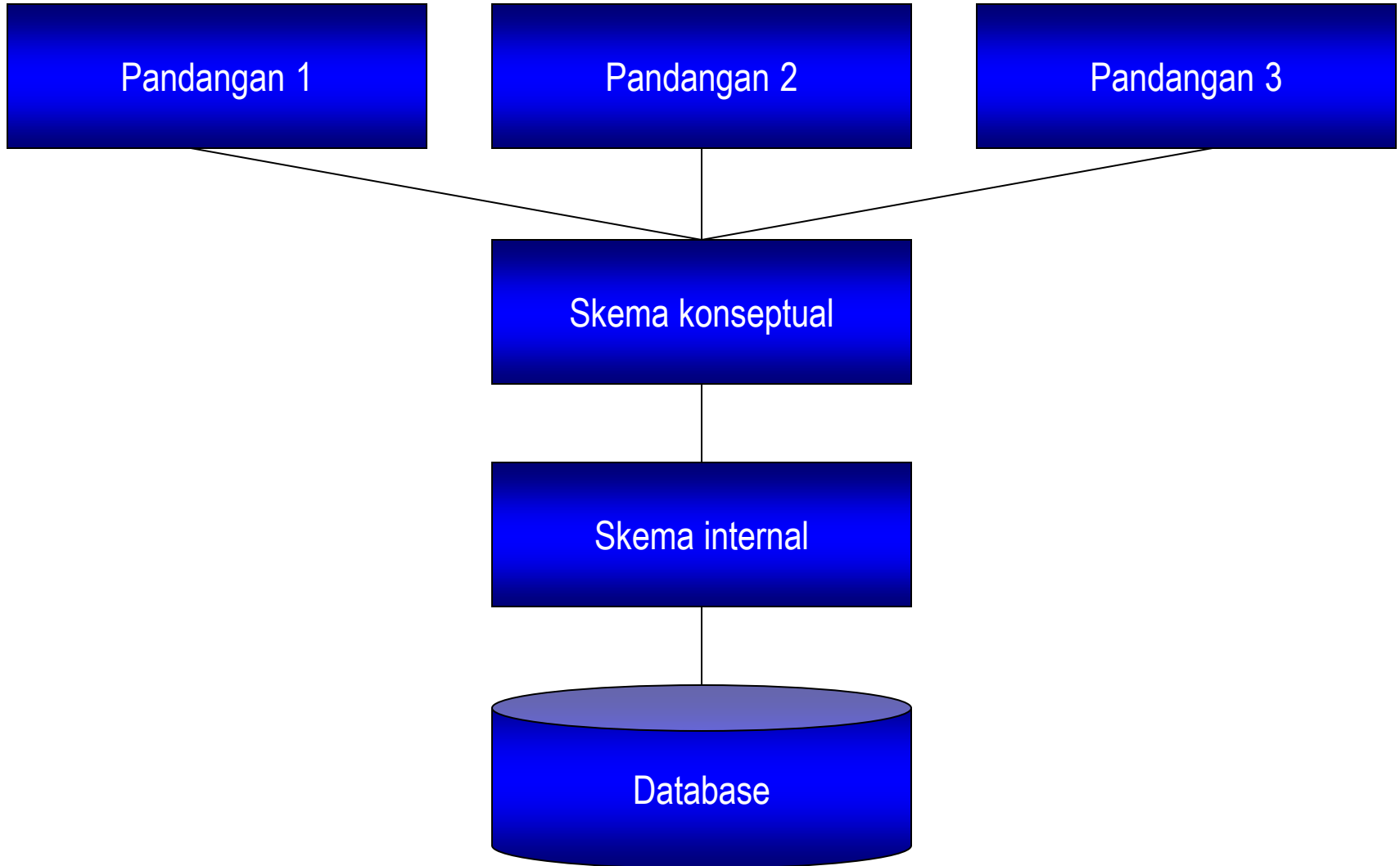
- ***End User*** : orang yang pekerjaannya memerlukan akses terhadap basis data untuk keperluan *query, update, generate report*.
- *End user* dibagi menjadi 4 kelompok, yaitu :
 - *Casual end users*
 - Mengakses basis data secara kadang-kadang tetapi mungkin memerlukan informasi yang berbeda untuk setiap kalinya.
 - Menggunakan bahasa *query* yang rumit dalam menspesifikasi *query*.
 - *Naive/Parametric end users*
 - Biasanya secara berkala melakukan *query* dan *update* basis data dengan menggunakan jenis *query & update* yang standar (transaksi yang telah di program dan di tes).
 - *Sophisticated end users*
 - Meliputi *engineers, scientists* dan *business analysts* – yang telah mengenal dengan baik dan menyeluruh mengenai fasilitas-fasilitas DBMS untuk memenuhi kebutuhan-kebutuhan yang kompleks.
 - *Stand-alone users*
 - Mereka yang memelihara basis data personal dengan menggunakan paket-paket program yang telah dibuat dan menyediakan menu-menu yang mudah untuk digunakan.

- *System analyst dan application programmers*
 - ***System Analyst*** : bertugas mendefinisikan kebutuhan-kebutuhan *end user* (khususnya *naive end user*) dan mengembangkan spesifikasi untuk transaksi-transaksi yang memenuhi keinginannya.
 - ***Applications Programmers*** : bertugas mengimplementasikan spesifikasi menjadi program (yang telah di tes secara intensif).

Arsitektur *Database*

- Arsitektur *database* dimaksudkan untuk membuat abstraksi terhadap *database*.
- Tujuannya agar DBMS dapat diakses secara efisien tanpa mengharuskan pemakai tahu secara detail tentang cara data disimpan dan dipelihara.
- ANSI-SPARC (*American National Standard Institute–Standards Planning and Requirement Comitte*) mendefinisikan 3 level dalam arsitektur *database*, yaitu :
 1. Level *eksternal*
 2. Level konseptual dan
 3. Level *internal*.

Arsitektur ANSI - SPARC



- *Internal level (internal schema)*
 - Menjelaskan struktur penyimpanan fisik dari basis data
 - Menggunakan model data fisik
- *Conceptual Level (conceptual schema)*
 - Menjelaskan struktur dari keseluruhan basis data untuk dipakai oleh satu komunitas *user*
 - Model data tingkat tinggi atau model data implementasi
- *External atau View Level (external schema atau user view)*
 - Menjelaskan sebagian basis data yang menjadi kebutuhan dari sekelompok *user* tertentu
 - Model data tingkat tinggi atau implementasi

Arsitektur Sistem Basis Data

- Sistem Tunggal (*Standalone*)
- Sistem Terpusat (*Centralized system*)
- Sistem *Client-server*
- Sistem Terdistribusi (Tersebar)

Sistem Tunggal (Standalone)

- DBMS, basis data dan aplikasi basis data ditempatkan pada komputer yang sama.
- Hanya bisa dipakai oleh satu pemakai pada saat yang bersamaan.
- Arsitektur sistem yang paling sederhana dan murah.
- Misal :diterapkan pada sebuah apotik kecil untuk menangani penjualan obat.

Sistem Terpusat

- Terdiri dari sebuah *server* dan sejumlah terminal.
- Yang terpusat adalah basis data, DBMS dan aplikasi basis data.

Sistem terpusat, terdapat dua bentuk :

1. *Server* berfungsi sebagai server DBMS dan aplikasi, terminalnya disebut *dumb* terminal (terminal pasif).
2. *Server* hanya berfungsi sebagai pengelola *file* basis data (*File server*), sedangkan DBMS dan aplikasi basis data ditempatkan pada setiap terminal kerja (*workstation*).

Sistem Client–Server

- Ditujukan untuk mengatasi kelemahan yang terdapat pada sistem terpusat. Terdiri dari 2 komponen utama yaitu *client* dan *server*.

Client yang menjalankan aplikasi basis data, *Server* yang menjalankan DBMS dan basis data.

Sistem *Client-Server* ada dua macam :

1. Arsitektur 2 lapis (2-tier)
2. Arsitektur 3 lapis (3-tier)

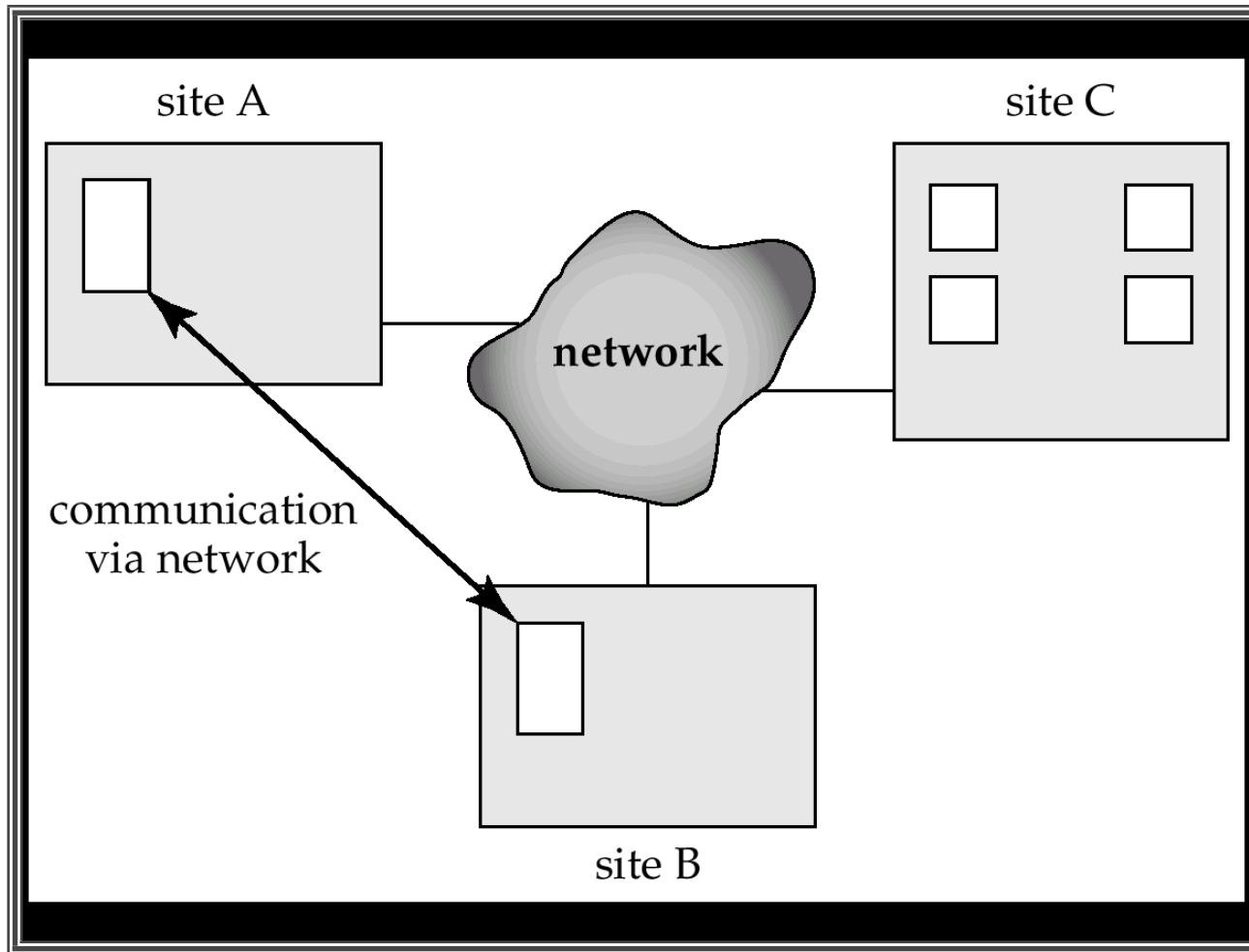
Sistem Client–Server

- Fungsi basis data terbagi dua bagian yaitu :
 1. *Front End* : Terdiri dari fasilitas antarmuka atau *interface*, seperti *form*, *grafis*, *sql user interface*
 2. *Back End* : Mengatur struktur akses dan optimasi *query*

Sistem Terdistribusi

- Sekumpulan data yang secara logis milik satu sistem, tetapi secara fisik tersebar.
- Basis data disimpan di beberapa komputer.
- Komputer pada sistem terdistribusi berkomunikasi satu sama lain.
- Komputer pada sistem terdistribusi disebut *site* atau *node*.

Sistem Terdistribusi



Sistem Terdistribusi

Alasan dibuatnya sistem terdistribusi :

1. **Data bersama**, pengguna pada *site* bisa menggunakan data pada *site* yang lain, misal seperti sistem perbankan, masing-masing cabang menyimpan data yang berhubungan dengan cabang tersebut.
2. **Ketersedian**, jika satu *site* gagal pada suatu sistem terdistribusi, *site* lain bisa melanjutkan operasi.
3. **Otonomi**, masing-masing administrator bisa mengontrol data yang disimpan sesuai dengan wewenangnya.

Data *Independence*

- **Tujuan utama** dari arsitektur basis data adalah **memelihara kemandirian data** (*data independence*).
- Artinya, perubahan pada satu level tidak mempengaruhi level yang lain.
- Metode **mengubah pola data dari isi data tersebut dan cara menyimpannya**, sehingga perubahan tersebut tidak menyebabkan suatu program aplikasi ditulis kembali.

Level Data *Independence*

Ada dua level data *independence*:

Physical data independence

- Kemampuan dalam merubah skema *internal* tanpa harus merubah skema konseptual.

Logical data independence

- Kemampuan dalam merubah skema konseptual tanpa harus mengubah skema *eksternal* atau program aplikasi. Jadi, *user* dapat merubah skema konseptual untuk menambah *database* (menambah *tipe record*), untuk merubah *constraint*, atau mengurangi *database* (menghapus *record*).
- Independensi data logis lebih sulit dicapai daripada independensi fisik karena program aplikasi sangat tergantung dengan struktur logika data yang diakses.

Data *Independence*

Prinsip ini harus diterapkan dalam pengelolaan sistem basis data dengan alasan:

1. DBA dapat mengubah isi, lokasi, perwujudan dalam organisasi basis data tanpa mengganggu program aplikasi yang sudah ada.
2. Pabrik atau *software* pengelolaan data dapat memperkenalkan produk-produk baru tanpa mengganggu program aplikasi yang sudah ada.
3. Untuk memindahkan perkembangan program aplikasi.
4. Memberikan fasilitas pengontrolan terpusat oleh DBA.

Mengenai MySQL

- MySQL adalah program *database server* yang mampu menerima dan mengirimkan datanya dengan cepat, *multiuser* serta menggunakan perintah standar SQL(*Structure Query Language*).
- MySQL adalah sebuah *database* yang dapat digunakan baik sebagai *client* maupun *server*.
- *Database* MySQL adalah sebuah perangkat lunak *database* yang berbentuk relasional atau dalam bahasa basis data disebut sebagai *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan SQL.



TERIMA KASIH