



12

## Database Client-Server

by: Ahmad Syauqi Ahsan

# Pemrograman Basis Data Client-Server

## □ Brief History

- ▣ Istilah client server pertama digunakan pada tahun 1980-an dalam bentuk PC yang terhubung dalam lingkungan jaringan.
- ▣ Secara actual model client/server dimulai akhir tahun 1980an.

## □ Pengertian Client Server

- ▣ Client : Suatu mesin/komputer yang selalu melakukan request kepada satu atau lebih mesin/komputer lain yang bertindak sebagai server.
- ▣ Server : Suatu mesin/komputer yang bertindak sebagai provider dari service yang memberikan response kepada komputer client

# Karakteristik Client-Server

- **Karakteristik Client**
  - ▣ Initiates requests
  - ▣ Waits for and receives replies
  - ▣ Usually connects to a small number of servers at one time
  - ▣ Typically interacts directly with end-users using a graphical user interface
  
- **Karakteristik Server**
  - ▣ Never initiates requests or activities
  - ▣ Listens to network and responds only to requests from connected, authorized clients
  - ▣ Waits for and replies to requests from connected clients
  - ▣ A server can remotely install/uninstall applications and transfer data to clients

# Keuntungan Client-Server

4

- Semua data tersimpan di server:
  - ▣ Sehingga memiliki tingkat keamanan dan kontrol yang lebih baik.
  - ▣ Kontrol akses dan sumber daya yang lebih baik, dapat memastikan hanya client/user yang berhak dapat mengakses atau merubah data
- Penyimpanan data terpusat:
  - ▣ Update data jauh lebih mudah untuk diatur dibandingkan dengan P2P (peer-to-peer).
  - ▣ Pada P2P architecture, data yang telah update kemungkinan harus didistribusikan kesemua client dalam network, hal ini membutuhkan waktu dan rawan kesalahan.

# Kekurangan Client-Server

5

- Traffic congestion, apabila terdapat banyak client yang mengakses server secara simultan dapat mengakibatkan overloadnya server.
- Apabila terjadi kerusakan pada server maka client tidak akan mendapat layanan dari server.

# Arsitektur Client-Server

6

- Mainframe
- File Sharing
- Client-Server
- Two-Tier
- Three-Tier
- Multi-Tier

# Mainframe

7

- ❑ Terdapat sebuah komputer pusat (host) yang memiliki sumber daya yang sangat besar, baik memori, processor maupun media penyimpanan.
- ❑ Melalui komputer terminal, pengguna mengakses sumber daya tersebut.
- ❑ Komputer terminal hanya memiliki monitor/keyboard dan tidak memiliki CPU.
- ❑ Semua sumber daya yang diperlukan terminal dilayani oleh komputer host.
- ❑ Model ini mulai berkembang pada akhir tahun 1980-an.

# File Sharing

- Server menyediakan file-file yang tersimpan di media penyimpanan untuk dapat diakses oleh pengguna.
- Arsitektur file sharing memiliki keterbatasan, terutama jika jumlah pengakses semakin banyak serta ukuran file yang di shaing sangat besar. Hal ini dapat mengakibatkan transfer data menjadi lambat.
- Model ini populer pada tahun 1990-an.  
Contoh : DBase, Microsoft Access



# Arsitektur Software Client-Server

- Karena keterbatasan sistem file sharing, dikembangkanlah arsitektur client/server.
- Salah satu hasilnya yaitu berupa software database server yang menggantikan software database berbasis file server. Dikenalkan pula RDBMS (Relational Database Management System).
- Dengan arsitektur ini, query data ke server dapat terlayani dengan lebih cepat karena yang ditransfer bukanlah file, tetapi hanyalah hasil dari query tersebut. RPC (Remote Procedure Calls) memegang peranan penting pada arsitektur client/server.

# Two-Tier

- Dikembangkan dari arsitektur software file server di era 1980 an.
- Dikembangkan untuk mendukung user-friendly interface.
- Arsitektur two-tier meningkatkan scalability dengan mengakomodir sampai dengan 100 user (arsitektur file-server hanya mengakomodir sekitar 12 user), dan meningkatkan fleksibilitas dengan cara mengizinkan data untuk di-share. Arsitektur ini biasanya berada dalam lingkungan yang homogen.
- Arsitektur two-tier membutuhkan intervensi operator yang minimal dan sering digunakan pada sistem pemrosesan informasi yang sederhana dan tidak kritis (non-critical).

# Two-Tier(2)

11

- TwoTier memiliki 3 komponen yang didistribusikan pada 2 layer:
  1. User System Interface
    - Seperti session, text input, dialog, dan display management services.
  2. Processing Management
    - Seperti process development, process enactment, process monitoring, dan process resource services
  3. Database Management
    - seperti data dan file services

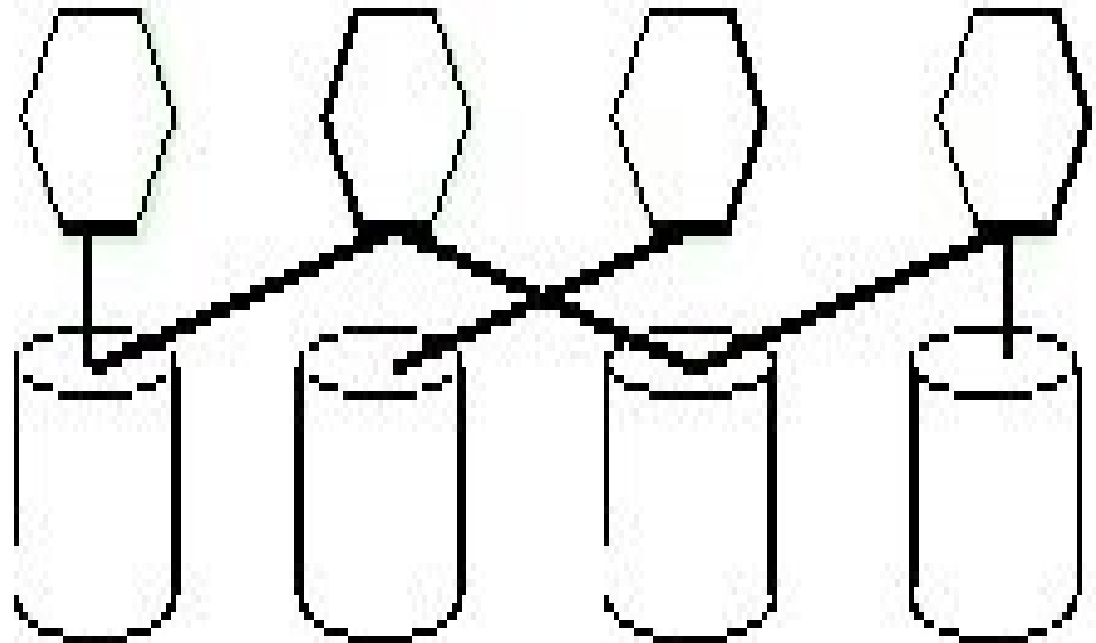
# Two-Tier(3)

12

## Two Tiers

User System Interface  
+ Some Processing  
Management

Database Management  
+ Some Processing  
Management



# Two-Tier: Petunjuk Penggunaan

- Two tier software architectures are used extensively in non-time critical information processing where management and operations of the system are not complex.
- This design is used frequently in decision support systems where the transaction load is light. Two tier software architectures require minimal operator intervention.
- The two tier architecture works well in relatively homogeneous environments with processing rules (business rules) that do not change very often and when workgroup size is expected to be fewer than 100 users, such as in small businesses.

# Two-Tier: Cost Limitation

- Scalability.
  - The two tier design will scale-up to service 100 users on a network. It appears that beyond this number of users, the performance capacity is exceeded. This is because the client and server exchange "keep alive" messages continuously, even when no work is being done, thereby saturating the network.
  - Implementing business logic in stored procedures can limit scalability because as more application logic is moved to the database management server, the need for processing power grows. Each client uses the server to execute some part of its application code, and this will ultimately reduce the number of users that can be accommodated.
- Batch jobs.
  - The two tiered architecture is not effective running batch programs. The client is typically tied up until the batch job finishes, even if the job executes on the server; thus, the batch job and client users are negatively affected

# Two-Tier: Cost Limitation (2)

- Interoperability.
  - The two tier architecture limits interoperability by using stored procedures to implement complex processing logic (such as managing distributed database integrity).
  - This means that to change or interoperate with more than one type of database management system, applications may need to be rewritten. Moreover, database management system's proprietary languages are generally not as capable as standard programming languages in that they do not provide a robust programming environment with testing and debugging, version control, and library management capabilities.
- System administration and configuration
  - Two tier architectures can be difficult to administer and maintain because when applications reside on the client, every upgrade must be delivered, installed, and tested on each client. The typical lack of uniformity in the client configurations and lack of control over subsequent configuration changes increase administrative workload.

# Three-Tier

16

- Diperkenalkan mulai tahun 1990 an untuk menanggulangi keterbatasan arsitektur two tier.
- Memisahkan layer Process management (middle tier) yang menanggung beban pemrosesan business logic & rules, pada server tersendiri.
- Beban database server (data/resource tier/database management) berkurang.
- Beban pada client yang biasanya menanggung presentation layer beserta process management berkurang.
- Client tidak dapat langsung berinteraksi dengan Database Server.
- Client dibatasi pada pemanggilan procedure atau object pada middle tier.



# Three-Tier (2)

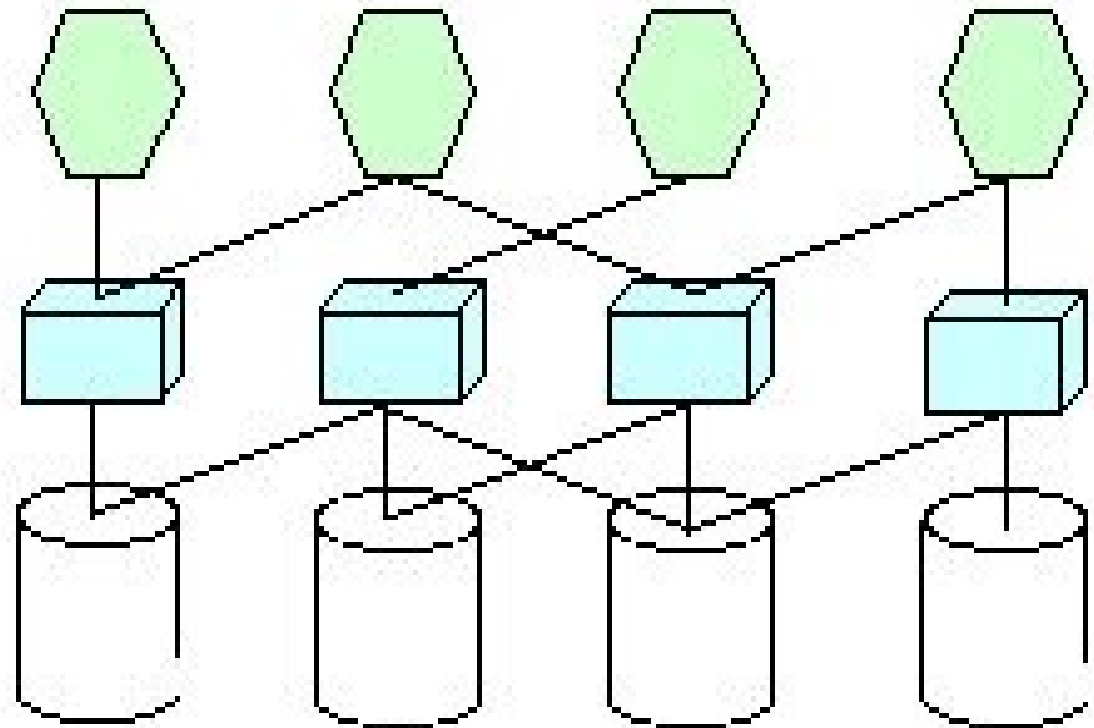
17

## Three Tiers

User System Interface

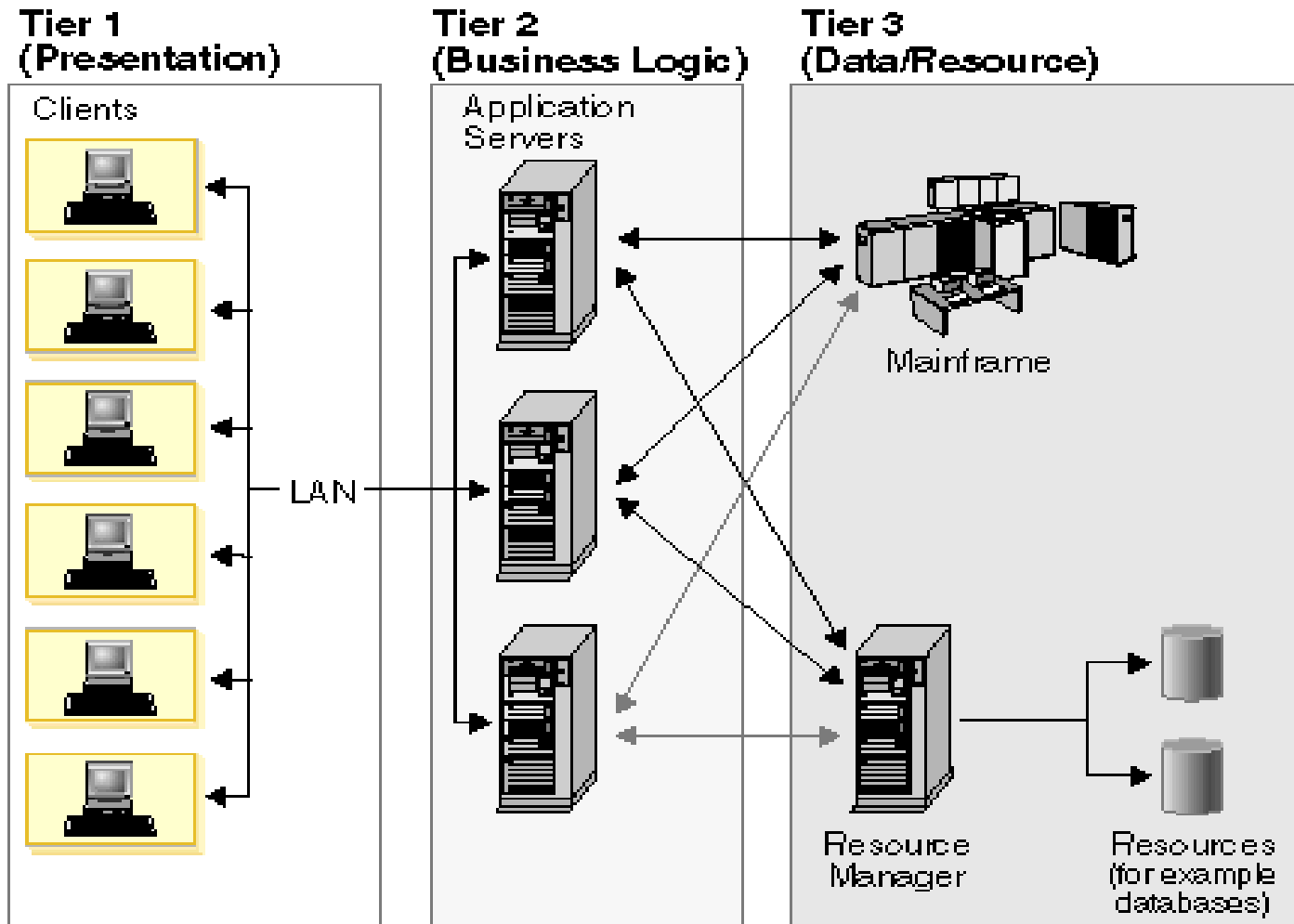
Process Management

Database Management



# Three-Tier (3)

18



# Three-Tier (4)

- Three Tiers memiliki 3 komponen yang didistribusikan pada 3 layer:
  1. User System Interface  
(such as session, text input, dialog, and display management services)
  2. Processing Management  
(such as process development, process enactment, process monitoring, and process resource services)
  3. Database Management  
(such as data and file services)

# Three Tier: Kapan Menggunakannya?

- Saat ini arsitektur 3-tier menjadi lebih populer dibandingkan dengan arsitektur 2-tier. Namun arsitektur 2-tier tetap tidak dapat ditinggalkan sepenuhnya, karena masih terdapat banyak aplikasi yang ideal menggunakan arsitektur 2-tier.
- Lalu bagaimana kita mengetahui model apa yang cocok untuk aplikasi kita dan sesuai dengan karakteristik perusahaan? Berikut ini adalah beberapa kriteria yang bisa digunakan untuk menentukan kapan sebuah aplikasi perlu menggunakan model 3-tier.

# Three Tier: Kapan Menggunakannya? (2)

- Jumlah layanan atau class aplikasi lebih dari 50
- Program aplikasi di buat atau ditulis dalam beberapa bahasa pemrograman yang berbeda untuk masing-masing organisasi.
- Dua atau lebih data source yang heterogen seperti dua DBMs yang berbeda atau DBMS dan file system
- Suatu aplikasi akan digunakan lebih dari 3 tahun. Apalagi jika kita akan merencanakan banyak modifikasi atau penambahan
- Beban kerja yang sangat tinggi. Lebih dari 50000 transaksi perhari
- Lebih dari 100 user yang secara bersamaan mengakses ke sistem dan database yang sama.
- Ekspektasi bahwa aplikasi akan terus berkembang sepanjang waktu

# Tanya Jawab

Terima Kasih