



# Sistem Basis Data

## ARSITEKTUR SISTEM

Alif Finandhita, S.Kom, M.T

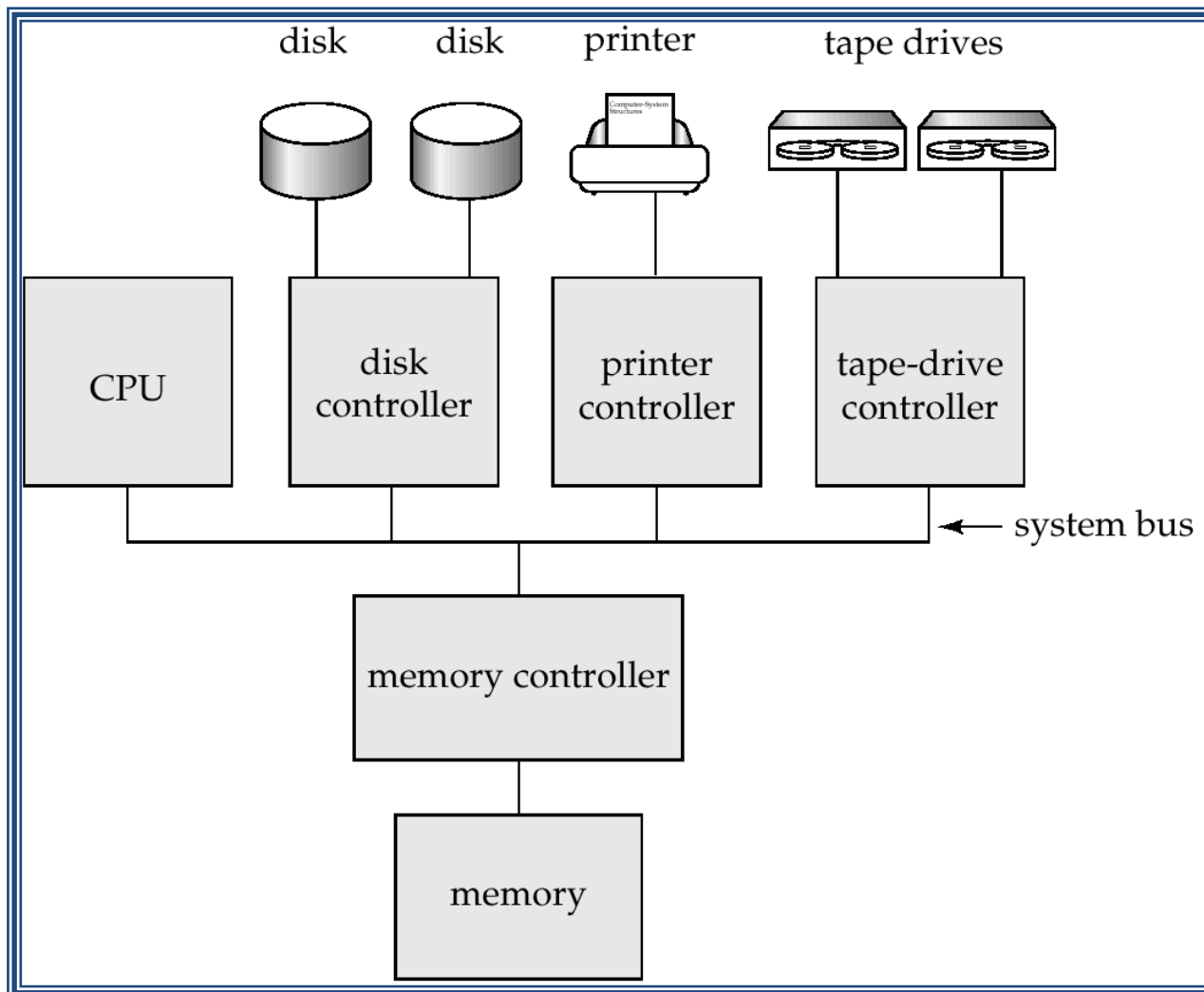
# ARSITEKTUR SISTEM

- Sistem Terpusat (*Centralized Systems*)
- Sistem *Client Server* (*Client-Server Systems*)
- Sistem Server (*Server Systems*)
- Sistem Paralel (*Parallel Systems*)
- Sistem Terdistribusi (*Distributed Systems*)

# Sistem Terpusat (*Centralized Systems*)

- Sistem berjalan pada satu sistem komputer tunggal dan tidak berinteraksi dengan sistem komputer lainnya
- Sistem ini meliputi sistem basis data yang berjalan pada *Personal Computer (PC)* hingga sistem basis data berkinerja tinggi yang berjalan pada sistem server (*High – End Server System*)
- Merupakan sistem komputer modern untuk tujuan umum yang terdiri atas satu hingga beberapa CPU dan banyak *device controller* yang dihubungkan melalui bus yang menyediakan akses ke memori bersama (*shared memory*)

# Sistem Terpusat (*Centralized Systems*) - 2



# Sistem Terpusat (*Centralized Systems*) - 3

- Sistem terpusat dapat dibedakan menjadi dua :
  - Single – User System
  - Multi – User System

# Sistem Terpusat *Single - User*

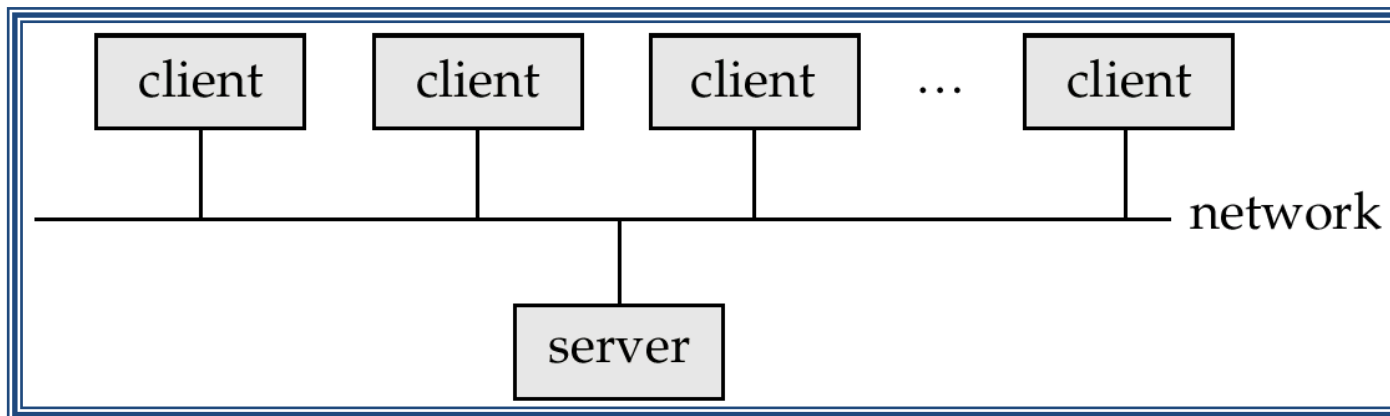
- Digunakan hanya oleh satu user saja, hanya satu CPU dan satu atau dua hard disk serta hanya satu user yang menggunakan komputer pada waktu yang bersamaan.
- Sistem operasi yang digunakan mungkin hanya mendukung untuk penggunaan satu user saja.
- Tidak menyediakan banyak fasilitas untuk kontrol konkurensi (*concurrency control*) dan pemulihan terhadap *system crash* karena hanya diakses oleh satu user saja.
- Contoh : *personal computer* atau *workstation*

# Sistem Terpusat *Multi - User*

- Memiliki kapasitas disk dan memori yang lebih banyak, jumlah CPU yang lebih banyak, serta sebuah sistem operasi yang mendukung *multi – user*.
- Sistem *multi – user* melayani banyak pengguna yang dihubungkan ke sistem melalui terminal.
- Sistem *multi – user* biasa disebut dengan *Server System*.
- Contoh : *Server Computer*

# Sistem *Client - Server*

- Pada arsitektur *client – server*, PC menggantikan peran terminal yang terhubung dengan sistem terpusat (*server*).
- PC dianggap mempunyai kemampuan sebagai antarmuka (*interface*) pengguna yang sering ditangani langsung oleh sistem terpusat.
- Sistem server melayani permintaan dari sejumlah sistem client yang digambarkan sebagai berikut :



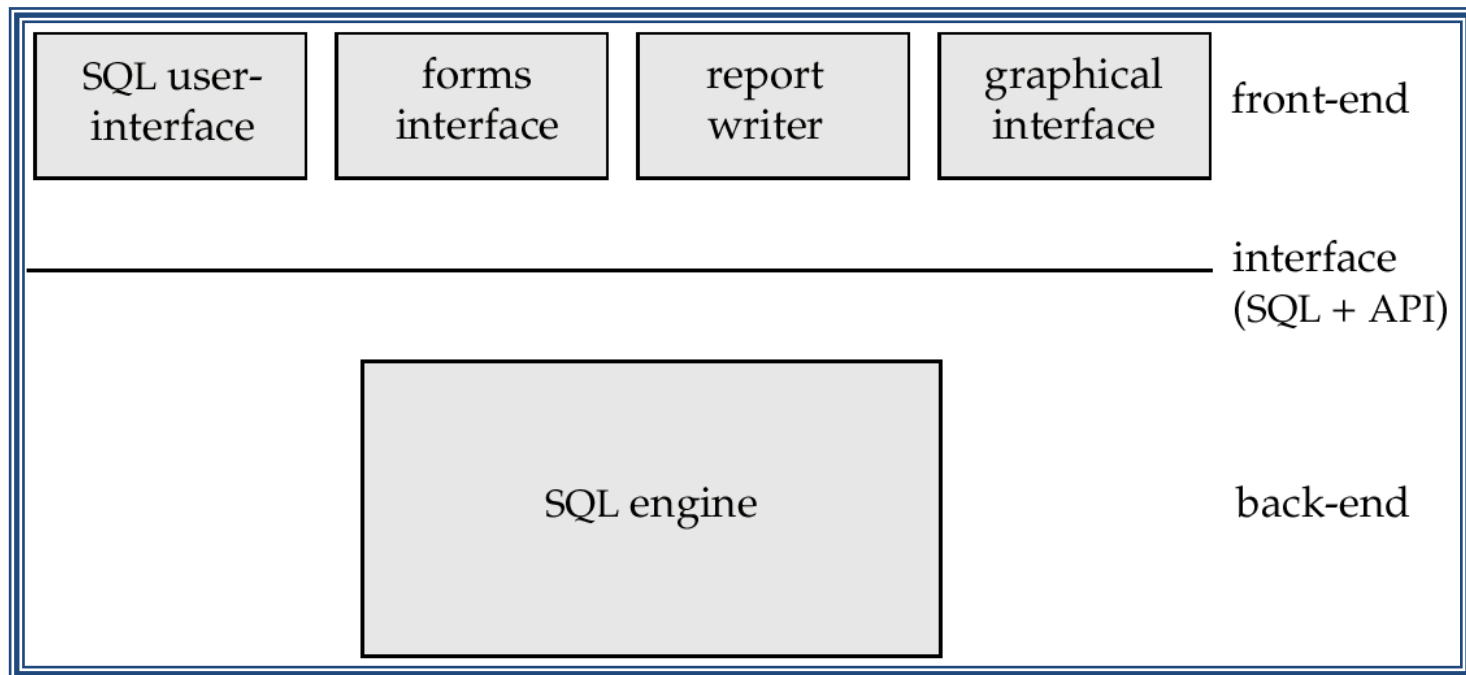


# Sistem *Client – Server* (2)

- Pada sistem client – server , fungsi basis data dapat dibagi menjadi dua bagian :
  - Back End  
Mengatur struktur akses evaluasi dan optimasi query, kontrol konkurensi, dan pemulihan (*recovery*).
  - Front End  
Terdiri atas tool – tool seperti fasilitas *form*, *report writers* dan tampilan antarmuka.

# Sistem *Client – Server* (3)

- *Interface* database antara *front-end* dan *back-end* bisa berupa DBMS (*SQL*) atau melalui program aplikasi tertentu



# Sistem *Client – Server* (4)

- Pada Sistem Basis Data generasi awal, pengembang perangkat lunak yang sama harus menyediakan *front end* dan *back end* sekaligus.
- Standar semacam ODBC (*Open Data Base Connectivity*) dan JDBC (*Java Data Base Connectiviey*) dibuat untuk menghubungkan client dengan server.
- Setiap client yang menggunakan ODBC atau JDBC dapat berhubungan dengan sembarang server yang menyediakannya

# Sistem *Client – Server* (5)

- Beberapa keuntungan mengganti jaringan mainframe dengan *workstation* atau PC yang terhubung dengan *back end* yang ada di komputer server :
  - Lebih efisien dari segi pembiayaan
  - Fleksibilitas dalam menempatkan sumber daya dan ketika akan menambah fasilitasnya
  - Antarmuka pengguna yang lebih baik
  - Lebih mudah dalam pemeliharaan

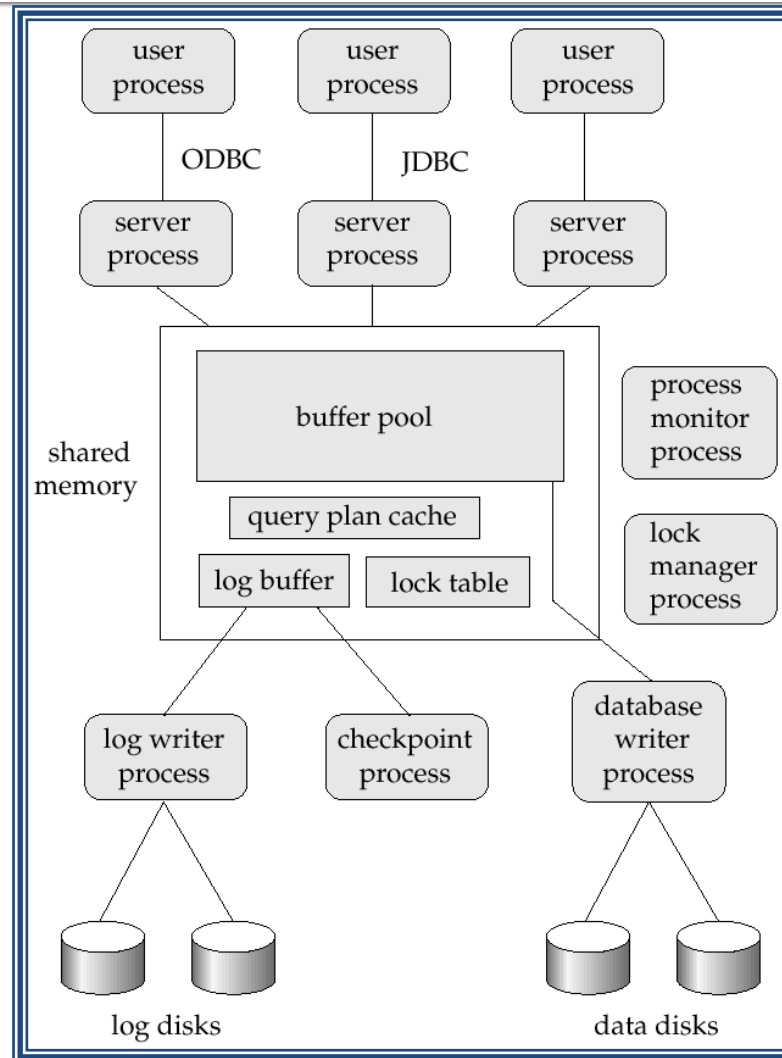
# Sistem Server

- Sistem server dapat dikelompokkan menjadi dua bagian :
  - Sistem Server Transaksi (*Transaction Servers*)
  - Sistem Server Data (*Data Server*)

# Sistem Server *Transaction Servers*

- *Transaction Servers* disebut juga sebagai sistem server query (*query server*).
- Menyediakan antarmuka dimana dengan antarmuka tersebut client dapat mengirimkan transaksi ke sistem server dimana transaksi tersebut dieksekusi dan hasilnya dikirim kembali ke client yang berwenang untuk menampilkan data.
- Permintaan transaksi bisa dilakukan langsung dengan menggunakan DBMS (SQL) atau melalui antarmuka program aplikasi.
- Sistem *transaction servers* terdiri atas banyak proses yang mengakses data pada *shared memory*.

# Struktur Proses *Transaction Servers*



# Struktur Proses *Transaction Servers* (2)

- Server transaksi secara spesifik terdiri dari beberapa proses yang mengakses data yang ada di *shared memory*.
- Proses yang membentuk bagian dari sistem basis data adalah :
  - **Server Process**  
Merupakan proses penerimaan query (transaksi) dari pengguna, mengeksekusinya, dan kemudian mengirimkan hasilnya kembali. Dapat berupa proses *multithread* yang memperbolehkan sebuah proses tunggal untuk mengeksekusi beberapa query user secara bersama – sama (konkuren). Biasanya proses ini merupakan proses *multiple multithread server process*.



# Struktur Proses *Transaction Servers* (3)

- **Log Manager Process**  
Menerapkan fungsi pengaturan penguncian, termasuk di dalamnya adalah *lock grant*, *lock release* dan pendeteksian *deadlock*.
- **Database Writer Process**  
Proses pengiriman kembali block buffer yang sudah dimodifikasi ke disk secara terus menerus.
- **Log Writer Process**  
Proses di server menambahkan *log record* ke *log record buffer* dalam *shared memory*, dan jika *log record* dibutuhkan, mereka meminta *log writer process* untuk mengirimkan *log record* media penyimpanan yang lebih stabil (*Non Volatile Storage*).

# Struktur Proses *Transaction Servers* (4)

- **Checkpoint Process**

Proses di server melakukan checkpoint secara teratur dan berkala.

- **Monitor Process**

Proses di server yang melakukan monitoring proses lain, jika ada proses yang gagal maka akan dilakukan aksi pemulihan untuk proses tersebut seperti pembatalan transaksi yang sedang dieksekusi, lalu kemudian mengulang proses tersebut.

# Struktur Proses *Transaction Servers* (5)

- Shared memory mengandung seluruh data yang digunakan bersama seperti :
  - Buffer Pool
  - Lock Table
  - Log Buffer, yang mengandung *log record* yang menunggu untuk dikirimkan ke *log* pada media penyimpanan yang lebih stabil
  - Cached Query Plans, digunakan kembali jika ada query yang telah di-*submit*, kemudian query tersebut di-*submit* lagi di waktu yang lain.

# Struktur Proses *Transaction Servers* (6)

- Sistem Server digunakan pada LAN dimana ada koneksi antara client dan server dengan akses kecepatan yang tinggi.
- Komputer server bisa mengirimkan data ke komputer client agar melakukan semua pemrosesan pada komputer client, lalu mengirimkan datanya kembali ke komputer server.
- Sistem server biasanya digunakan pada sistem basis data berorientasi objek.

# Sistem Paralel (*Parallel Systems*)

- Sistem ini menggunakan beberapa prosesor dan disk yang dirancang untuk dijalankan secara paralel - apabila dimungkinkan - selama hal tersebut digunakan untuk memperbaiki kinerja dari DBMS.
- Sistem paralel di jalankan oleh berbagai multi prosesor.
- Sistem paralel menghubungkan beberapa mesin yang berukuran kecil untuk menghasilkan keluaran sebuah mesin yang berukuran besar dengan skalabilitas yang lebih besar dan keandalan dari basis datanya.
- Untuk menopang beberapa prosesor dengan akses yang sama pada satu basis data, sistem DMBS paralel harus menyediakan manajemen sumber daya yang dapat diakses bersama. Sumber daya yang digunakan tergantung dari aplikasi atau lingkungan yang digunakan.

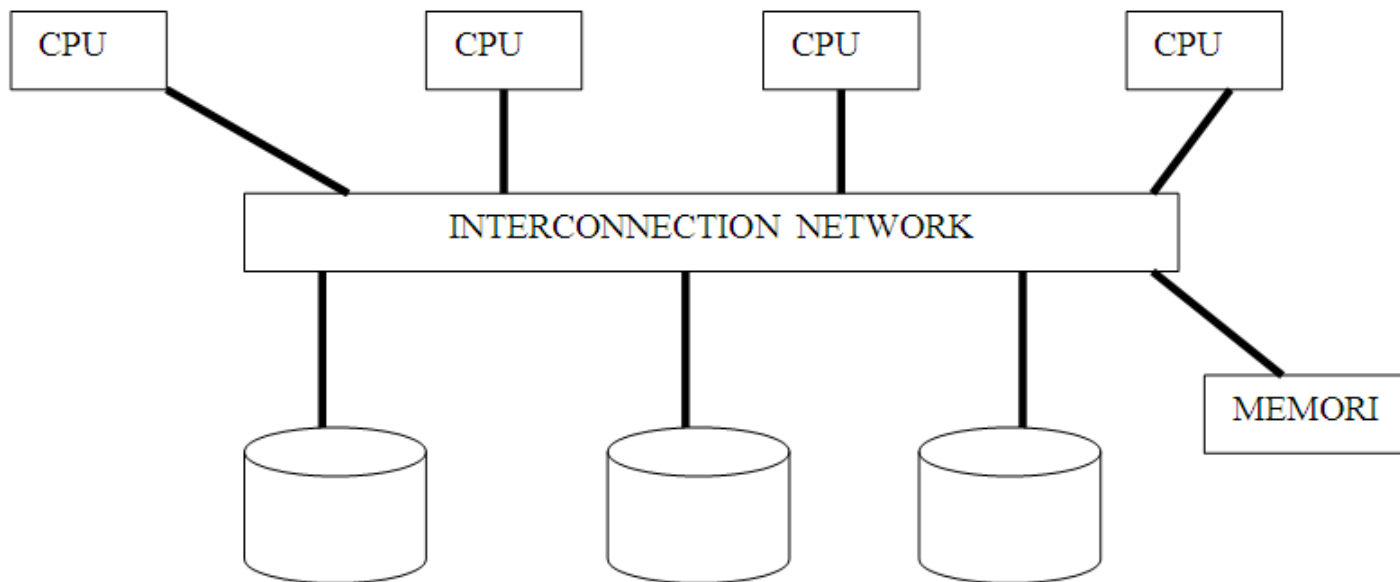
# Sistem Paralel (*Parallel Systems*) – 2

- Teknologi paralel ini biasanya digunakan untuk basis data yang berukuran sangat besar (terabites ) atau sistem yang memproses ribuan transaksi perdetik.
- Paralel DBMS dapat menggunakan arsitektur yang diinginkan untuk memperbaiki kinerja yang kompleks untuk mengeksekusi query dengan menggunakan paralel scan, join dan teknik sort yang memperbolehkan node dari banyak prosesor untuk menggunakan bersama pemrosesan kerja yang digunakan.

# Sistem Paralel (*Parallel Systems*) – 3

- Arsitektur yang digunakan pada Sistem DBMS Paralel :
  - Penggunaan memory bersama (*share memory*)
  - Penggunaan disk bersama (*share disk*)
  - Penggunaan secara sendiri-sendiri (*share nothing*)

# Sistem Paralel (*Parallel Systems*) – *Shared Memory*



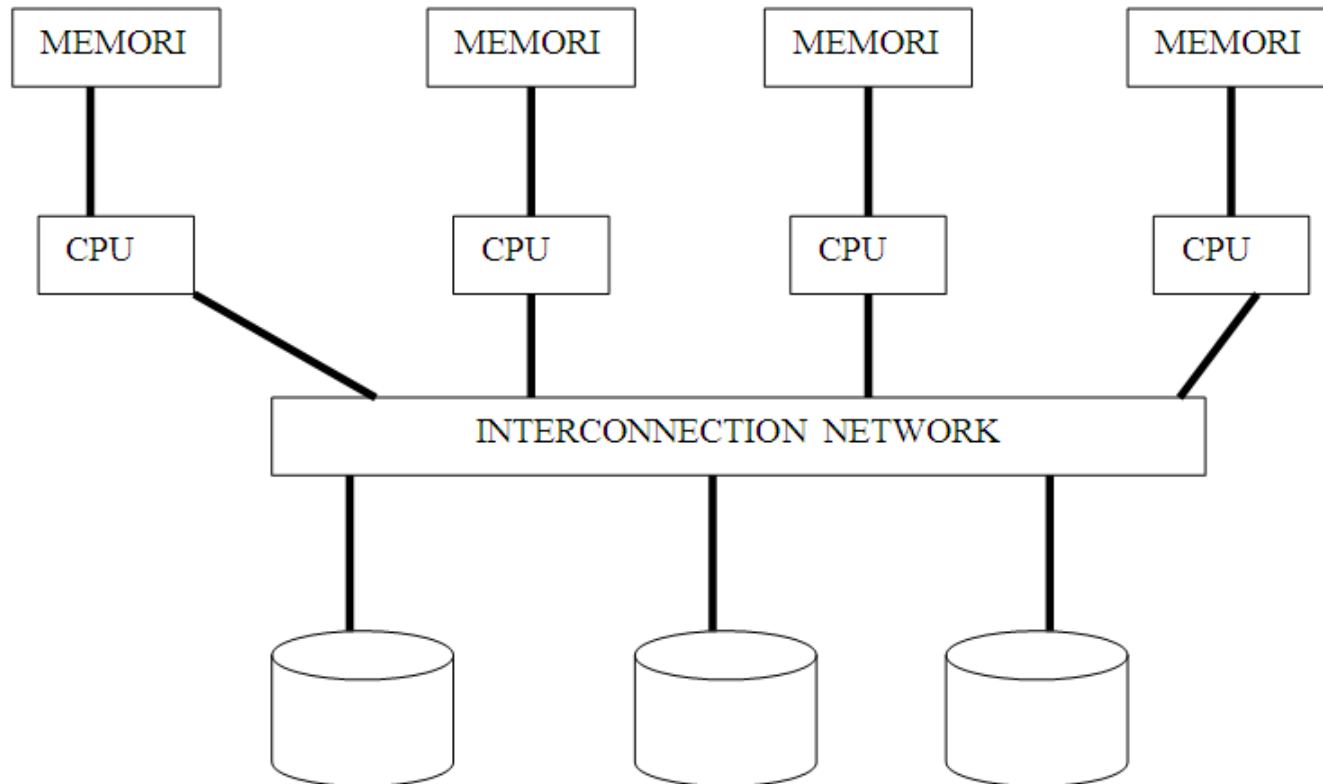
Arsitektur Sistem Paralel dengan Penggunaan Memori Bersama  
(*Shared Memory*)



# Sistem Paralel (*Parallel Systems*) – *Shared Memory* (2)

- Sebuah arsitektur yang menghubungkan beberapa prosesor di dalam sistem tunggal yang menggunakan memori secara bersama – sama.
- Dikenal dengan nama *SMP (Symmetric Multiprocessing)*, metode ini sering digunakan dalam bentuk *workstation* personal yang mensupport beberapa mikroprosesor dalam paralel DBMS, *RISC (Reduced Instruction Set Computer)* yang besar berbasis mesin sampai bentuk mainframe yang besar.
- Arsitektur ini menghasilkan pengaksesan data yang sangat cepat yang dibatasi oleh beberapa prosesor .

# Sistem Paralel (*Parallel Systems*) – *Shared Disk*

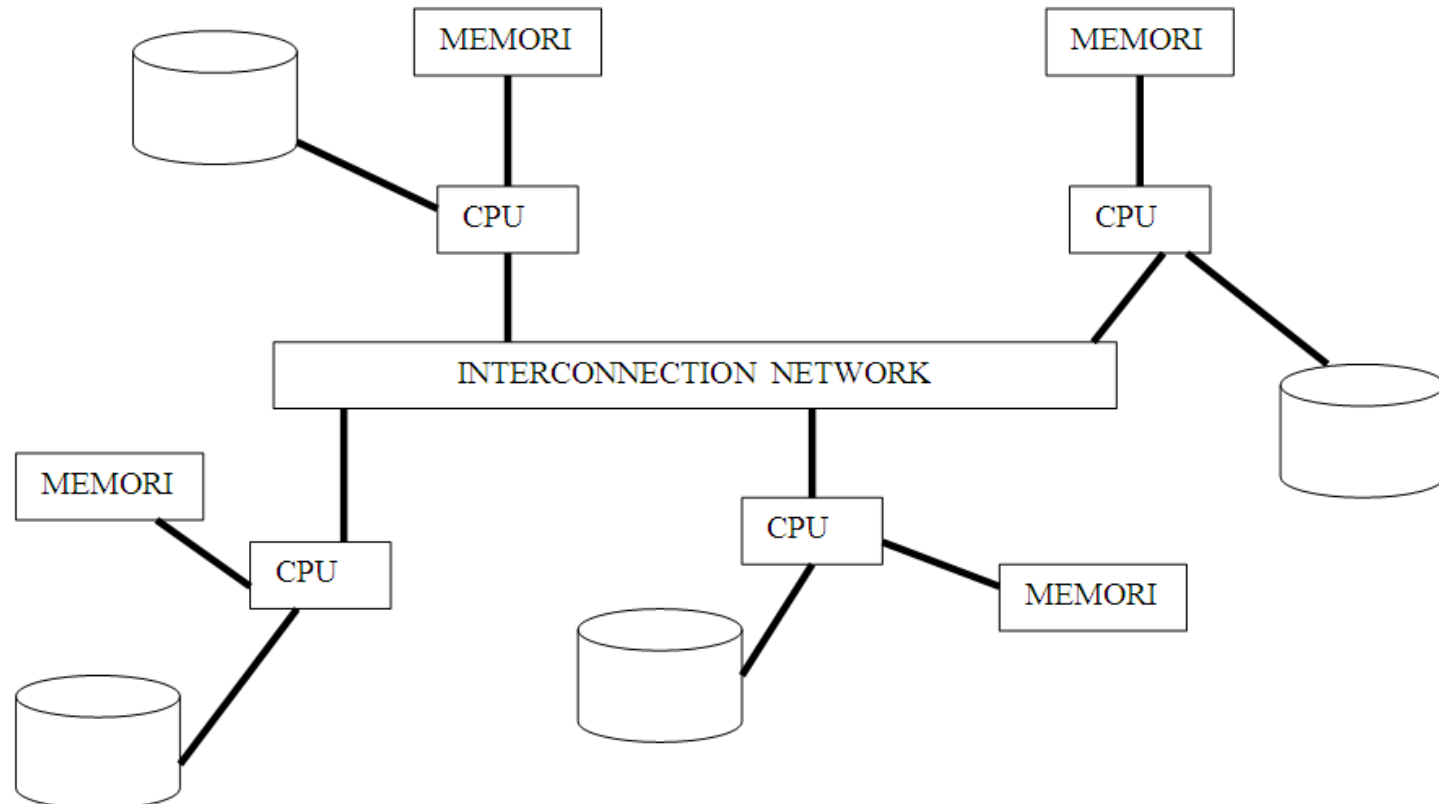


Arsitektur Sistem Paralel dengan Penggunaan Disk Bersama (*Shared Disk*)

# Sistem Paralel (*Parallel Systems*) – *Shared Disk (2)*

- Sebuah arsitektur yang mengoptimalkan jalannya suatu aplikasi yang tersentralisasi dan membutuhkan keberadaan data dan kinerja yang tinggi.
- Setiap prosesor dapat mengakses langsung semua disk, tetapi prosesor tersebut memiliki memorinya sendiri – sendiri.
- Arsitektur ini menghapus masalah pada penggunaan memori bersama tanpa harus mengetahui sebuah basis data di partisi. Arsitektur ini dikenal dengan **cluster**.

# Sistem Paralel (*Parallel Systems*) – *Shared Nothing*



Arsitektur Sistem Paralel dengan Penggunaan Sendiri - sendiri  
(*Shared Nothing*)

# Sistem Paralel (*Parallel Systems*) – *Shared Nothing* (2)

- Sering di kenal dengan Massively parallel processing (MPP) yaitu arsitektur dari beberapa prosesor di mana setiap prosesor adalah bagian dari sistem yang lengkap, yang memiliki memori dan disk.
- Basis data ini di partisi untuk semua disk pada masing – masing sistem yang berhubungan dengan basis data dan data di berikan secara transparan untuk semua pengguna yang menggunakan sistem .
- Arsitektur ini lebih dapat di hitung skalabilitasnya dibandingkan dengan shared memory dan dapat dengan mudah mensupport prosesor yang berukuran besar. Kinerja dapat optimal jika data di simpan di lokal DBMS.

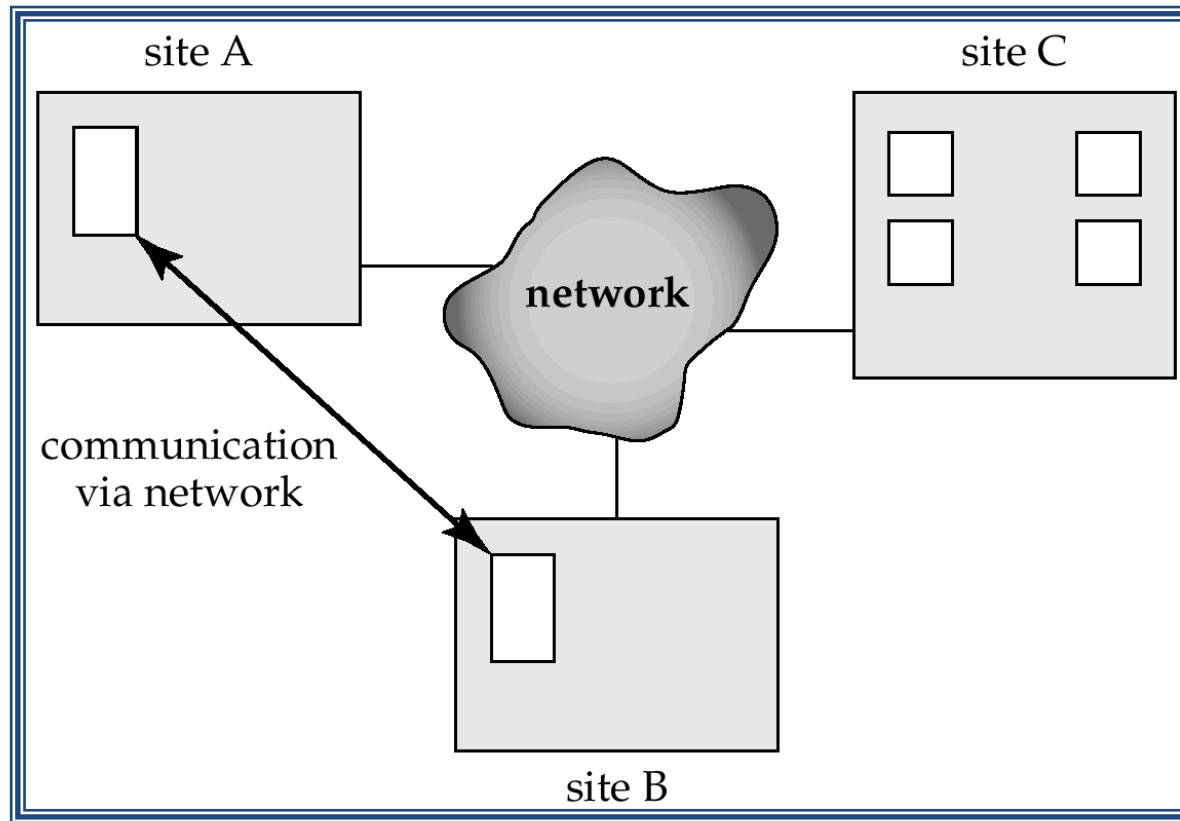
# Sistem Paralel (*Parallel Systems*) – *Shared Nothing* (3)

- Arsitektur pada penggunaan secara sendiri – sendiri (*shared nothing*) hampir sama dengan DBMS terdistribusi, namun pendistribusian data pada paralel DBMS hanya berbasis pada kinerjanya saja.
- Node pada DDBMS adalah merupakan pendistribusian secara geographic, administrasi yang terpisah, dan jaringan komunikasi yang lambat, sedangkan node pada paralel DBMS adalah hubungan dengan komputer yang sama atau site yang sama.

# Sistem Terdistribusi

- Sekumpulan data yang **secara logis** adalah milik satu sistem yang sama, tetapi **secara fisik** tersebar pada beberapa tempat di jaringan.
- Basis Data disimpan di beberapa komputer.
- Komputer pada sistem terdistribusi berkomunikasi dengan satu sama lain.
- Tidak saling berbagi memori atau disk.
- Komputer pada sistem terdistribusi disebut sebagai *site* atau *node*.

# Sistem Terdistribusi (2)





# Sistem Terdistribusi (3)

- Sistem Basis Data terdistribusi berisi sekumpulan site dimana di tiap – tiap site dapat berpartisipasi dalam pengekseskuan transaksi yang mengakses data pada satu atau beberapa site.
- Sistem Basis Data terdistribusi sering terpisah secara geografis.

# Sistem Terdistribusi (4)

Jenis transaksi dalam sistem terdistribusi :

- Transaksi Lokal  
Transaksi yang hanya mengakses data hanya dari site dimana transaksi dilakukan
- Transaksi Global  
Transaksi yang mengakses data pada beberapa site yang berbeda

# Sistem Terdistribusi (5)

Alasan membuat sistem terdistribusi :

- **Data Bersama**

Pengguna pada satu site bisa menggunakan data yang ada pada site lain. Misalnya pada sistem perbankan, dimana masing – masing cabang menyimpan data yang berhubungan dengan cabang tersebut.

- **Ketersediaan**

Jika satu site gagal pada sistem terdistribusi, maka site lainnya dapat melanjutkan operasi sampai selesai.

# Sistem Terdistribusi (6)

## ■ Otonomi

- Masing – masing site bisa mengontrol data yang disimpan secara lokal.
- Administrator global bertanggung jawab terhadap keseluruhan sistem, sedangkan administrator lokal bertanggung jawab hanya pada site masing – masing yang dikelolanya.
- Masing – masing administrator lokal bisa memiliki otonomi lokal yang berbeda tergantung pada rancangan sistem basis data terdistribusi yang digunakan.

# Sistem Terdistribusi (7)

Tipe Sistem Terdistribusi :

- Homogen

**Semua** site pada satu sistem terdistribusi memiliki DBMS dan software komunikasi yang **sama**.

- Heterogen

Site pada satu sistem terdistribusi memiliki DBMS dan software komunikasi yang **berbeda** satu sama lain

# Sistem Terdistribusi (8)

Metode Sistem Terdistribusi :

- **Replikasi**

Data/tabel disalin pada sejumlah server yang berbeda .

- **Fragmentasi**

Data/tabel dipilah kemudian disebar ke sejumlah fragmen.

# Sistem Terdistribusi (9)

Keuntungan Sistem Terdistribusi :

- **Pengawasan distribusi dan pengambilan data**  
Seorang pemakai bisa mengakses data yang tersedia pada site lain.
- **Reliabilitas dan Ketersediaan**  
Dapat terus menerus berfungsi dalam menghadapi kegagalan site individu. Saat satu site gagal, site lain dapat melanjutkan operasi jika data telah direplikasi pada beberapa site.
- **Otonomi Lokal**  
Mengizinkan sekelompok individu untuk mengelola data mereka sendiri, mengurangi ketergantungan pada pusat.

# Sistem Terdistribusi (10)

Keuntungan Sistem Terdistribusi :

- **Kecepatan Pemrosesan Query**

Jika sebuah query melibatkan data pada beberapa site, maka site dapat membagi query ke dalam sub-query yang dapat dieksekusi dalam bentuk paralel.

- **Efisien dan Fleksibel**

Data dalam sistem terdistribusi dapat disimpan dekat dengan titik dimana data dipergunakan. Data dapat secara dinamis bergerak, disalin atau dapat dihapus salinannya.



# Sistem Terdistribusi (10)

Kerugian Sistem Terdistribusi :

- **Implementasi Sistem lebih mahal**
- **Kemungkinan Kesalahan Lebih Besar**  
Site dalam sistem terdistribusi beroperasi secara paralel sehingga lebih sulit menjamin kebenaran algoritmanya.
- **Biaya Pemrosesan Tinggi**  
Penambahan perhitungan dibutuhkan untuk mencapai koordinasi antar site. Dalam memilih sebuah rancangan sistem basis data, harus mengimbangi keuntungan dan kerugian basis data terdistribusi