

# MATAKULIAH BASIS DATA TERDISTRIBUSI

Rahmadden, M.Kom  
NIDN : 1007128301



Pertemuan 2 – Karakteristik, Tipe/Jenis, Arsitektur dan  
Topologi Basis Data Terdistribusi



## Visi Program Studi

Menjadi Program Studi Teknik Informatika yang unggul dalam bidang mobile computing di Sumatera tahun 2030

# #01

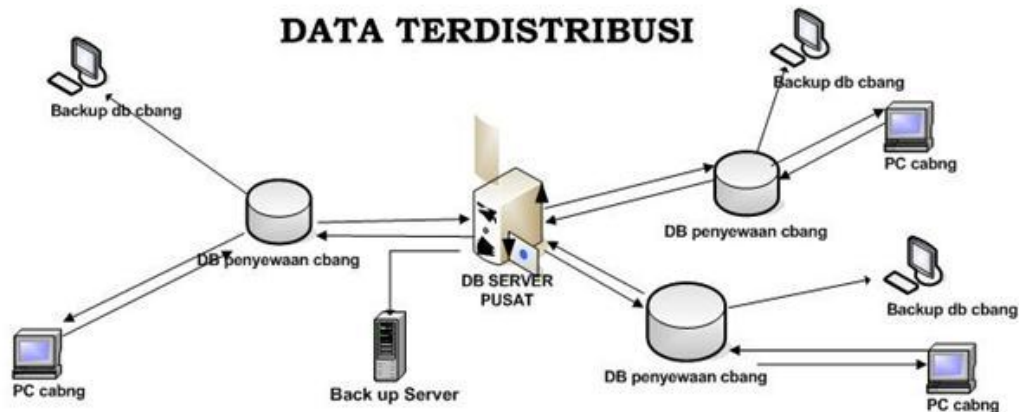
## KARAKTERISTIK BASIS DATA TERDISTRIBUSI





# Karakteristik Basis Data Terdistribusi

1. Data disimpan di sejumlah tempat
2. Processor pada tempat yang berbeda tersebut dihubungkan dengan jaringan computer
3. Sistem Basis Data Terdistribusi bukan terdiri dari sekumpulan file yang berada pada berbagai tempat tetapi pada sebuah basis data di berbagai tempat.
4. Setiap tempat secara mandiri memproses permintaan user yang membutuhkan akses ke data di tempat tersebut dan juga mampu untuk memproses data yang tersimpan di tempat lain.



# #02

## TIPE / JENIS BASIS DATA TERDISTRIBUSI



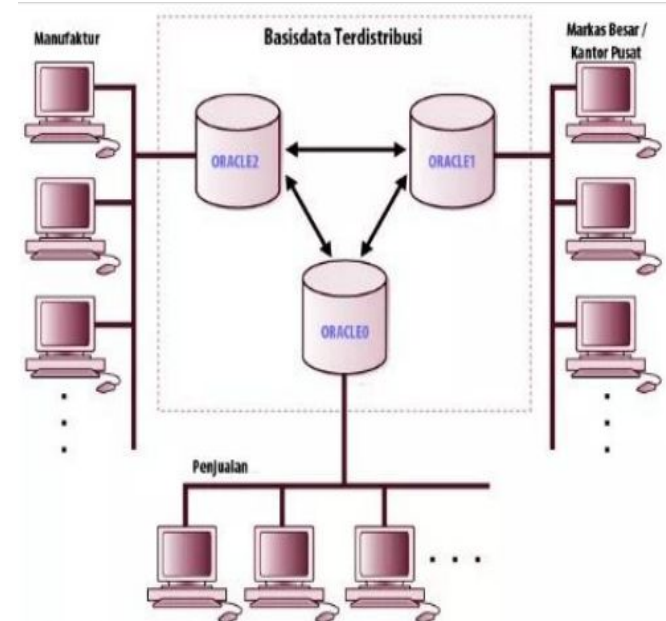


# Tipe Basis Data Terdistribusi

Terdapat dua tipe basis data terdistribusi :

a. Homogen □ yaitu sistem dimana setiap tempat menjalankan tipe DBMS yang sama

- Semua pihak/lokasi memiliki perangkat lunak yang identic
- Mereka mengenali satu sama lain dan setuju untuk bekerja sama dalam mengolah permintaan pengguna
- Setiap situs menyerahkan bagian dari otonomi dalam hal hak untuk mengubah skema atau perangkat lunak pengguna sebagai sistem tunggal





## Tipe Basis Data Terdistribusi

b. Heterogen □ yaitu sistem dimana setiap tempat yang berbeda menjalankan DBMS yang berbeda, baik Relational DBMS (RDBMS) atau non relational DBMS.

Dalam sistem database terdistribusi heterogen, setidaknya salah satu database menggunakan skema dan perangkat lunak yang berbeda.

Gambaran basis data terdistribusi yang heterogen dapat dilihat pada gambar dibawah ini :





## Tipe Basis Data Terdistribusi

Kunci keberhasilan membangun sistem yang heterogen adalah pada standarisasi protokol gateway.

Protokol gateway adalah **API (Application Programming Interface)** yang memungkinkan DBMS berfungsi untuk aplikasi eksternal, sebagai contoh seperti : ODBC dan JDBC.





# #03

## ARSITEKTUR BASIS DATA TERDISTRIBUSI





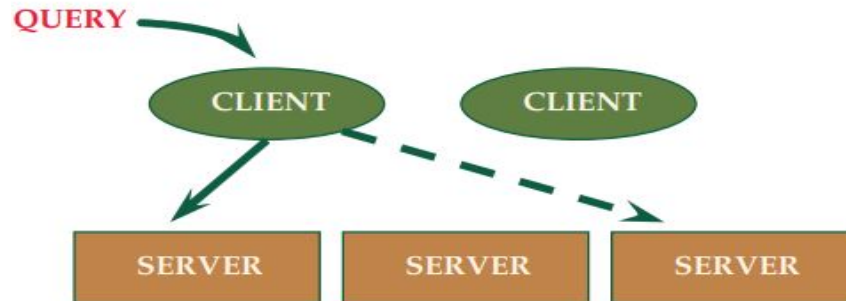
# ARSITEKTUR BASIS DATA TERDISTRIBUSI

Dua arsitektur alternatif DBMS terdistribusi adalah Client/Server dan Collaboration Server.

## 1. Client-Server

Sistem client-server mempunyai satu atau lebih proses client dan satu atau lebih proses server, dan sebuah proses client dapat mengirim query ke sembarang proses server.

Client bertanggung jawab pada antar muka untuk user, sedangkan server mengatur data dan mengeksekusi transaksi. Sehingga suatu proses client berjalan pada sebuah personal computer dan mengirim query ke sebuah server yang berjalan pada mainframe.





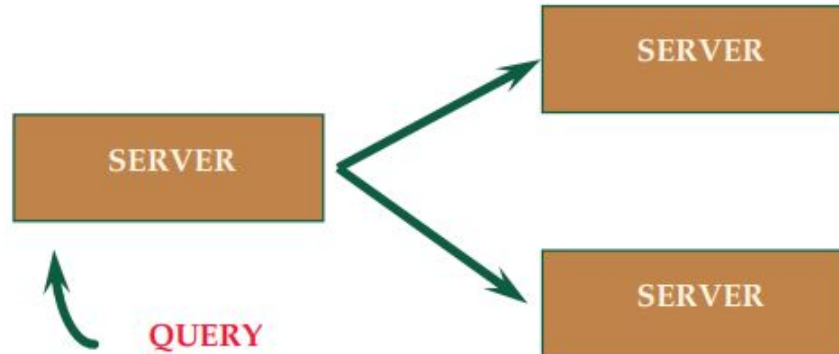
Arsitektur ini menjadi sangat populer untuk beberapa alasan yakni :

1. Implementasi yang relatif sederhana karena pembagian fungsi yang baik dan karena server tersentralisasi.
2. Mesin server yang mahal utilisasinya tidak terpengaruh pada interaksi pemakai, meskipun mesin client tidak mahal.
3. Pemakai dapat menjalankan antarmuka berbasis grafis sehingga pemakai lebih mudah dibandingkan antar muka pada server yang tidak user-friendly.



## 2. Collaboration Server

Proses client cukup kompleks dan terjadi overlap dengan server; sehingga perbedaan antara client dan server menjadi jelas. Untuk mengurangi perbedaan digunakan alternatif arsitektur client-server yaitu **sistem Collaboration Server**. Pada sistem ini terdapat sekumpulan server basis data, yang menjalankan transaksi data lokal yang bekerjasama mengeksekusi transaksi pada beberapa server. Jika server menerima query yang membutuhkan akses ke data pada server lain, sistem membangkitkan subquery yang dieksekusi server lain dan mengambil hasilnya bersama-sama untuk menggabungkan jawaban menjadi query asal.

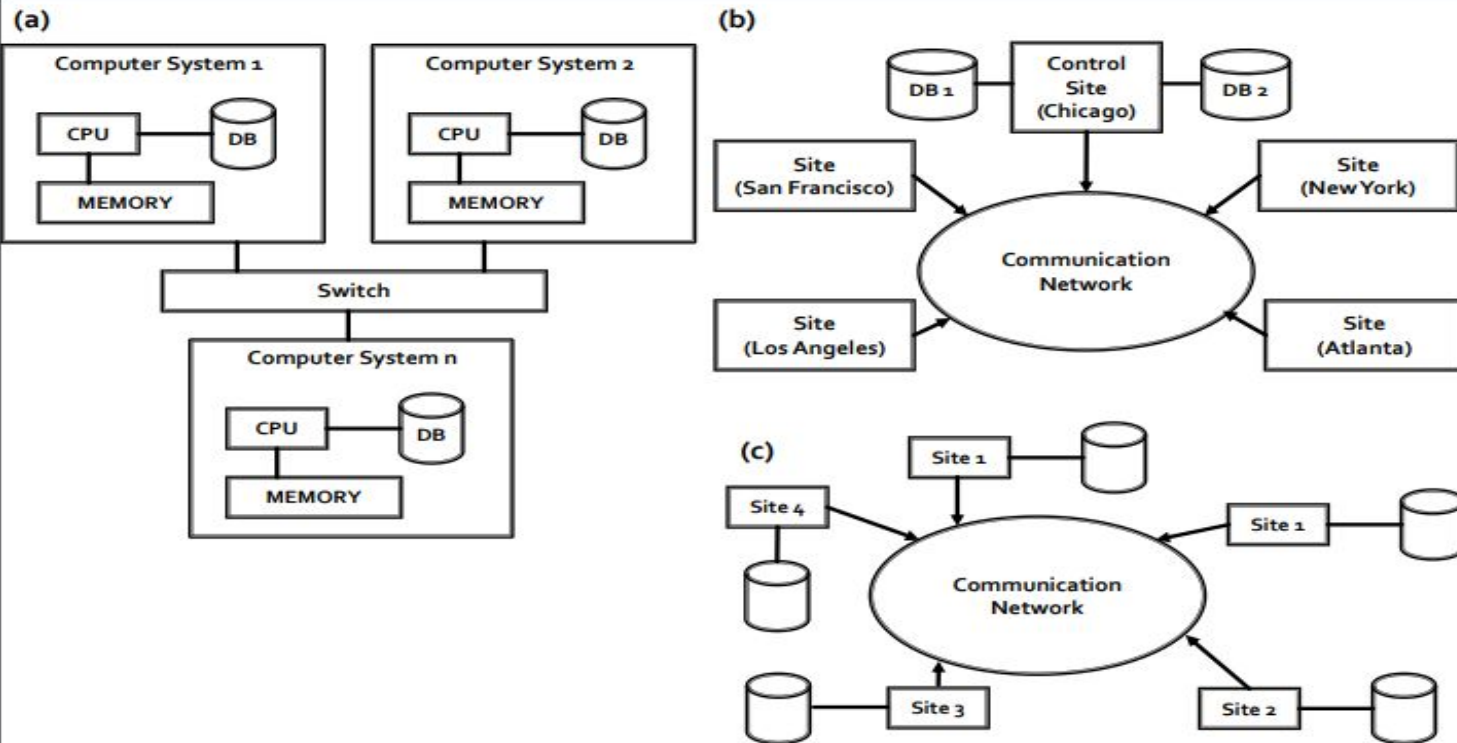




## 3. Sistem Middleware

Sistem middleware dirancang untuk memungkinkan satu query dijalankan pada multipel server dengan server-server basis data tidak mengelola eksekusi pada multi tempat.

Untuk koordinasi sub-query dan eksekusi join dilakukan oleh perangkat lunak tersendiri yang disebut middleware.



Some different database system architectures.

- (a) Shared Nothing architecture.
- (b) A Networked architecture with a centralized database at one of the sites.
- (c) A Truly distributed database architecture.

# #04

## TOPOLOGI BASIS DATA TERDISTRIBUSI





# TOPOLOGI / STRUKTUR BASIS DATA TERDISTRIBUSI

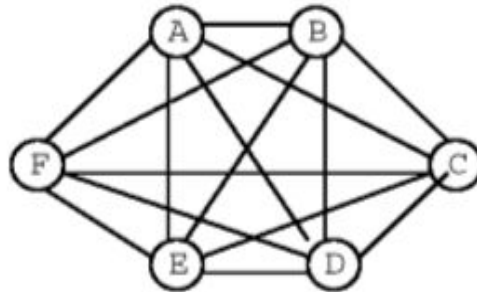
- A. Fully Connected Network
- B. Partially Connected Network
- C. Tree Structured Network
- D. Ring Network
- E. Star Network



## 1. Fully Connected Network :

Tiap site dalam *Fully Connected Network* terkoneksi secara langsung dengan situs lainnya. *Link* yang ada menjadi banyak dan menyebabkan biaya instalasi besar. Topologi jenis ini tidak praktis untuk diterapkan dalam sistem yang besar.

- Keuntungan □ kalau salah satu node rusak, yang lainnya masih dapat berjalan (tetapi biaya mahal).
- Kerugian □ control management tidak terjamin



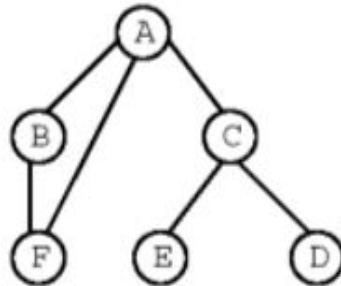
Fully connected network



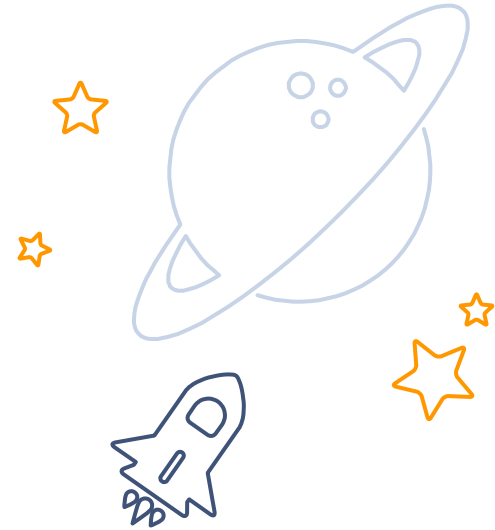
## 2. Partially Connected Network :

*Link* yang ada hanya antara beberapa situs sehingga biaya instalasi menjadi lebih rendah. Namun, biaya komunikasi bisa menjadi lebih mahal

- Keuntungan □ reliability rendah, biaya dapat ditekan
- Kerugian □ control management tidak terjamin



Partially connected network

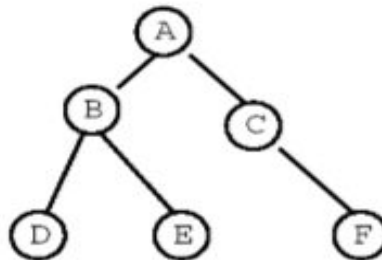


### 3. Tree Structure Network :

Biaya instalasi dan komunikasi pada topologi jenis ini biasanya rendah. Namun, jika terjadi *failure link* atau *failure site* maka pengaksesan data menjadi terhambat dan mengakibatkan availibilitas/ketersediaan menjadi rendah.

- Keuntungan □ bersifat sentral, control management lebih terjamin
- Kerugian □ kalau node pusat (A) rusak, semua akan rusak.

Cat : setiap proses dimulai dari bawah.



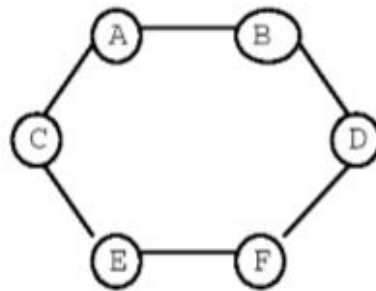
Tree structured network



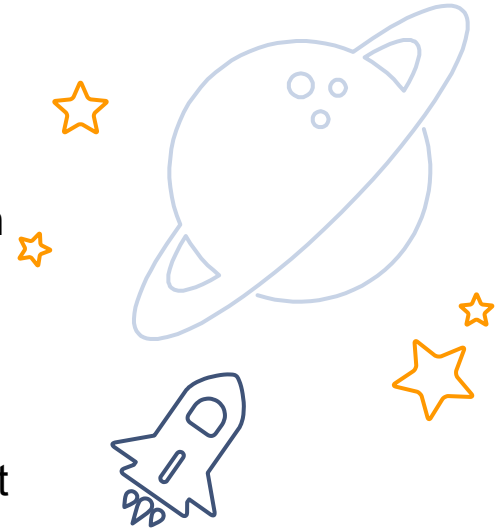
## 4. Ring Network (LAN) :

Biaya komunikasi tinggi karena jika ingin mengakses sebuah situs bisa jadi harus menempuh banyak *link*.

- Keuntungan □ rusak satu, yang lain masih berjalan
- Kerugian □ Control management kurang terjamin karena bersifat desentralisasi



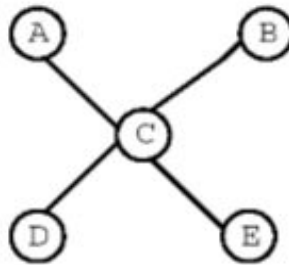
Ring network



## 5. Star Network (LAN) :

Biaya komunikasi rendah karena setiap situs paling banyak mengakses dua *link* ke situs lain.

- Keuntungan □ control management lebih terjamin, karena bersifat sentral - reliability rendah
- Kerugian □ kalau pusat rusak, yang lainnya rusak




Star network



← → ↺ 🏠 en.wikipedia.org/wiki/Apache\_Cassandra

📄 ☆ 📶

🌐 Apl 📧 Gmail 🗺️ Maps 📺 YouTube 🗣️ Pernyataan Switch-... 🌐 PAK Online 🌐 SIMDOS 🌐 Lembaga Layanan... 🌐 RDF Primer 🌐 18.04.1398\_jurnal\_e... 🌐 PKP Login | JAIA - Journ...



WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Current events  
Random article  
About Wikipedia  
Contact us  
Donate

Contribute

Help  
Learn to edit  
Community portal  
Recent changes  
Upload file


Tools

What links here  
Related changes  
Special pages  
Permanent link  
Page information  
Cite this page  
Wikidata item

Print/export

# Apache Cassandra

From Wikipedia, the free encyclopedia




This article **contains content that is written like an advertisement**. Please help improve it by removing promotional content and inappropriate external links, and by adding encyclopedic content written from a neutral point of view. (October 2019) (Learn how and when to remove this template message)

**Apache Cassandra** is a [free and open-source](#), [distributed](#), [wide column store](#), [NoSQL database](#) management system designed to handle large amounts of data across many [commodity servers](#), providing high availability with no [single point of failure](#). Cassandra offers robust support for [clusters](#) spanning multiple [datacenters](#),<sup>[2]</sup> with asynchronous masterless replication allowing low latency operations for all clients. Cassandra offers the distribution design of Amazon DynamoDB with the data model of Google's Bigtable.

**Contents** [hide]

- History
  - Releases
- Main features
  - Cassandra Query Language
  - Known issues
    - Tombstones
  - Data model
- Management and monitoring
- Notable applications
- See also
- References
- Bibliography

**Apache Cassandra**



**cassandra**

**Original author(s)** Avinash Lakshman, Prashant Malik / Facebook

**Developer(s)** Apache Software Foundation

**Initial release** July 2008; 12 years ago

**Stable release** 3.11.8 / August 31, 2020; 41 days ago<sup>[1]</sup>

**Repository** [Cassandra Repository](#)🔗

**Written in** Java

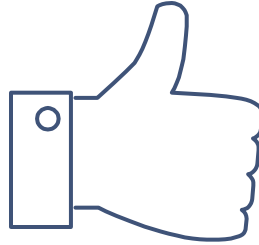
**Operating system** [Cross-platform](#)

Apache Cassandra adalah sistem manajemen open source database terdistribusi yang dirancang untuk menangani data yang sangat besar di beberapa server.









# THANKS!

Ada Pertanyaan?

Boleh juga ke

WA

FB

IG