

# **SISTEM TERDISTRIBUSI DAN TERPUSAT**

Disusun untuk memenuhi  
Laporan Sistem Paralel dan Terdistribusi

Oleh:

**ALIF MURSALIN**

**2208001010002**

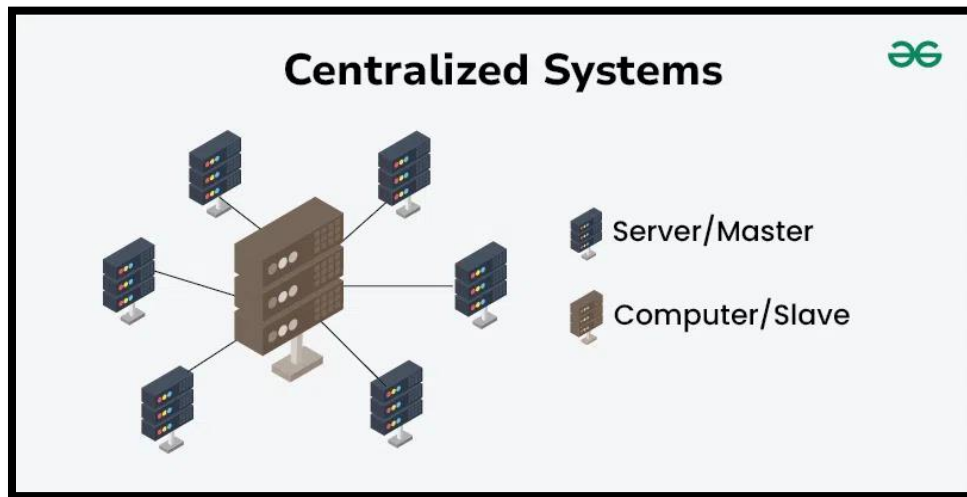


**PROGRAM STUDI DIII MANAJEMEN INFORMATIKA**  
**JURUSAN INFORMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS SYIAH KUALA**

## POINT 1

Perbedaan utama antara sistem terdistribusi dan sistem terpusat terletak pada bagaimana sumber daya dan pengolahan data dikelola serta didistribusikan dalam suatu jaringan atau organisasi. Berikut penjelasannya:

### 1.1. Sistem Terpusat



Sistem terpusat adalah sistem di mana semua komputasi dan pengolahan data dilakukan pada satu titik pusat, biasanya berupa server atau mainframe. Semua klien atau perangkat dalam jaringan akan mengirimkan permintaan dan data ke pusat ini untuk diproses.

#### Contoh Nyata:

- **Sistem Perbankan Kuno:** Dulu, banyak sistem perbankan menggunakan server pusat yang mengelola semua transaksi, dengan cabang bank dan ATM terhubung langsung ke server pusat untuk setiap proses.
- **Sistem ERP (Enterprise Resource Planning) Tradisional:** Sistem ERP yang berbasis server pusat, di mana seluruh data perusahaan diproses dan dikelola di satu server.

#### Kelebihan:

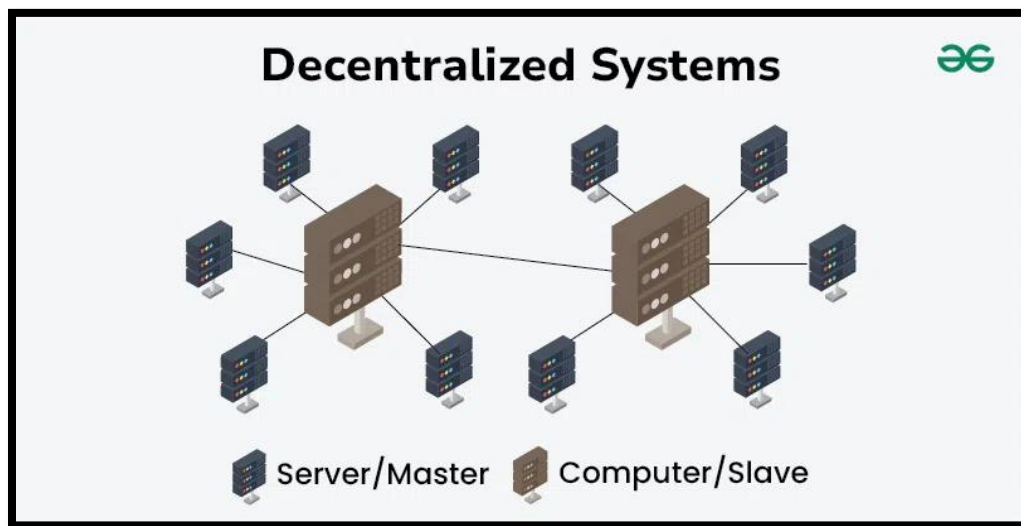
- **Kontrol Pusat:** Administrasi dan kontrol yang mudah karena semua aktivitas dikelola di satu tempat.

- **Keamanan Lebih Terkonsentrasi:** Lebih mudah mengelola keamanan data karena hanya ada satu titik utama yang perlu dijaga.
- **Manajemen Sumber Daya yang Efisien:** Semua sumber daya diatur dari satu server, mengurangi duplikasi.

#### Kekurangan:

- **Single Point of Failure (SPOF):** Jika server pusat mengalami gangguan atau kerusakan, seluruh sistem akan terhenti.
- **Beban pada Server:** Seluruh beban dipikul oleh server pusat, yang bisa menyebabkan penurunan performa jika permintaan meningkat drastis.
- **Kurangnya Skalabilitas:** Menambah kapasitas sering kali sulit karena tergantung pada kemampuan pusat.

## 1.2. Sistem Terdistribusi



Sistem terdistribusi adalah sistem di mana komputasi dan pengolahan data dibagi di antara beberapa titik atau node dalam jaringan. Setiap node dapat bekerja secara independen dan bersama-sama untuk menyelesaikan tugas yang lebih besar.

### Contoh Nyata:

- **Blockchain:** Teknologi blockchain adalah contoh sempurna dari sistem terdistribusi. Di sini, tidak ada server pusat; setiap node dalam jaringan memiliki salinan buku besar yang sama dan memverifikasi transaksi secara independen.
- **Sistem Cloud Modern:** Layanan seperti Google Drive atau AWS menggunakan server terdistribusi di beberapa lokasi untuk menyediakan layanan cloud yang dapat diakses dari mana saja.
- **CDN (Content Delivery Network):** Layanan seperti Cloudflare atau Akamai menggunakan server terdistribusi di seluruh dunia untuk mendistribusikan konten secara lebih cepat ke pengguna berdasarkan lokasi mereka.

### Kelebihan:

- **Toleransi Kesalahan (Fault Tolerance):** Jika salah satu node gagal, yang lainnya tetap bisa berfungsi, menjaga ketersediaan layanan.
- **Skalabilitas Tinggi:** Sistem ini mudah diperluas dengan menambahkan lebih banyak node atau server.
- **Desentralisasi:** Tidak ada satu titik pusat yang mengontrol semua, mengurangi ketergantungan pada satu entitas.

### Kekurangan:

- **Kompleksitas Manajemen:** Sistem ini lebih sulit dikelola karena melibatkan banyak node yang harus disinkronkan.
- **Keamanan yang Lebih Sulit:** Karena banyaknya titik koneksi, risiko keamanan lebih menyebar dan sulit dikontrol.

- **Konsistensi Data:** Dalam beberapa kasus, menjaga konsistensi data antar node bisa menjadi tantangan, terutama di sistem yang besar.

#### Perbandingan Singkat:

Aspek	Sistem Terpusat	Sistem Terdistribusi
<b>Pengelolaan Data</b>	Di satu server pusat	Dikelola oleh banyak node
<b>Keandalan</b>	Tergantung pada pusat (rentan SPOF)	Lebih andal karena toleransi terhadap kesalahan
<b>Keamanan</b>	Mudah dikelola di satu titik	Lebih sulit karena banyak titik
<b>Kinerja</b>	Bisa lambat jika beban pada pusat tinggi	Lebih cepat karena distribusi beban kerja
<b>Skalabilitas</b>	Terbatas oleh kapasitas pusat	Sangat skalabel
<b>Contoh</b>	Server mainframe perbankan, ERP	Blockchain, Cloud, CDN

Pemilihan antara sistem terdistribusi dan terpusat tergantung pada kebutuhan organisasi. Sistem terdistribusi lebih cocok untuk aplikasi yang membutuhkan ketersediaan tinggi dan fleksibilitas, sementara sistem terpusat bisa lebih optimal untuk pengelolaan yang terfokus dan sederhana.

## **B. POINT 2**

Bentuk **implementasi sistem terdistribusi** dalam kehidupan sehari-hari saat ini sangat beragam dan menyentuh hampir setiap aspek teknologi modern. Beberapa implementasi yang paling umum digunakan adalah dalam layanan cloud computing, blockchain, dan aplikasi berbasis jaringan seperti media sosial, streaming, serta komputasi paralel. Berikut adalah beberapa contoh implementasi sistem terdistribusi yang sering kita temui:

### **1. Cloud Computing**

Cloud computing adalah salah satu bentuk implementasi sistem terdistribusi yang paling banyak digunakan. Dalam cloud, layanan seperti penyimpanan, pengolahan data, dan aplikasi disebarkan melalui banyak server di berbagai lokasi yang dikelola oleh penyedia layanan cloud.

#### **Contoh:**

- **Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure:** Layanan ini menggunakan server yang tersebar di berbagai pusat data di seluruh dunia. Misalnya, saat menggunakan Google Drive, data kita tidak disimpan di satu lokasi fisik tetapi didistribusikan di berbagai server, yang memberikan redundansi dan keandalan lebih tinggi.

#### **Kelebihan:**

- **Skalabilitas:** Sumber daya dapat ditambah atau dikurangi sesuai kebutuhan.
- **Ketersediaan Tinggi:** Layanan tetap dapat diakses meskipun salah satu server mengalami kegagalan.
- **Efisiensi Biaya:** Pengguna hanya membayar sumber daya yang mereka gunakan.

#### **Penggunaan Sehari-Hari:**

- Penyimpanan dokumen di Google Drive atau Dropbox.
- Penggunaan aplikasi berbasis cloud seperti Google Docs atau Microsoft 365.

## 2. Content Delivery Network (CDN)

CDN adalah jaringan server yang terdistribusi di berbagai lokasi geografis untuk mengirimkan konten (seperti video, gambar, situs web) kepada pengguna berdasarkan lokasi terdekat. Ini adalah contoh jelas dari sistem terdistribusi yang bertujuan untuk mengurangi latensi dan meningkatkan kecepatan akses.

### Contoh:

- **Cloudflare, Akamai, Fastly:** Ketika kita mengakses situs web yang menggunakan CDN, konten dikirim dari server terdekat dengan lokasi geografis kita, sehingga waktu pemuatan menjadi lebih cepat.

### Kelebihan:

- **Akses Lebih Cepat:** Mengurangi waktu respon karena konten disimpan di server terdekat.
- **Reliabilitas:** Konten tetap tersedia meskipun salah satu server mengalami masalah.

### Penggunaan Sehari-Hari:

- Saat menonton video di YouTube atau Netflix, CDN memastikan video diputar dengan lancar tanpa buffering.
- Mengakses situs web seperti e-commerce (misalnya Amazon), yang menggunakan CDN untuk mempercepat pengiriman konten kepada pelanggan.

## 3. Blockchain

Blockchain adalah teknologi terdistribusi di mana data (transaksi, informasi) disimpan dalam blok-blok yang terhubung dan tersebar di berbagai node. Setiap node memiliki salinan lengkap dari blockchain, sehingga tidak ada satu otoritas pusat yang mengendalikan jaringan.

**Contoh:**

- **Bitcoin, Ethereum:** Teknologi di balik mata uang kripto ini adalah blockchain, yang memungkinkan transaksi dilakukan tanpa perantara (misalnya, bank), dengan setiap transaksi diverifikasi oleh node yang terdistribusi di seluruh dunia.

**Kelebihan:**

- **Desentralisasi:** Tidak ada entitas tunggal yang mengendalikan data, sehingga transparan dan aman dari manipulasi.
- **Keamanan Tinggi:** Data yang dimasukkan ke dalam blockchain tidak dapat diubah, menciptakan rekam jejak yang transparan.

**Penggunaan Sehari-Hari:**

- Penggunaan mata uang kripto untuk transaksi online.
- Penggunaan **smart contract** dalam platform seperti Ethereum untuk transaksi otomatis tanpa perantara.

**4. Sistem Peer-to-Peer (P2P)**

Sistem P2P adalah sistem di mana setiap komputer dalam jaringan bertindak sebagai server dan klien, berbagi sumber daya satu sama lain tanpa memerlukan server pusat. Ini adalah salah satu bentuk paling murni dari sistem terdistribusi.

**Contoh:**

- **BitTorrent:** Dalam jaringan P2P seperti BitTorrent, file diunduh dari beberapa sumber sekaligus, bukan dari satu server pusat. Setiap pengguna yang memiliki sebagian file bisa menjadi "seed" bagi pengguna lain.

**Kelebihan:**

- **Tidak Bergantung pada Pusat:** Ketersediaan konten tidak tergantung pada satu server.



- **Efisiensi Bandwidth:** File diunduh dari banyak sumber secara bersamaan, mempercepat proses unduhan.

#### **Penggunaan Sehari-Hari:**

- Saat berbagi file menggunakan aplikasi seperti uTorrent.
- Beberapa aplikasi pesan instan juga menggunakan arsitektur P2P untuk mengirimkan data secara langsung antara pengguna.

### **5. Aplikasi Berbasis Microservices**

Dalam arsitektur microservices, aplikasi besar dipecah menjadi komponen kecil yang saling berkomunikasi melalui API. Setiap layanan kecil dapat dijalankan di server yang berbeda, menjadikan sistem ini contoh sistem terdistribusi.

#### **Contoh:**

- **Netflix, Amazon:** Aplikasi besar seperti Netflix menggunakan arsitektur microservices, di mana setiap layanan, seperti streaming, rekomendasi, atau penagihan, dikelola oleh layanan yang terpisah. Jika satu layanan mengalami masalah, layanan lainnya tetap bisa berjalan.

#### **Kelebihan:**

- **Pemeliharaan Lebih Mudah:** Setiap layanan dapat dikelola secara terpisah, membuat pengembangan lebih fleksibel.
- **Skalabilitas:** Setiap layanan dapat diskalakan sesuai dengan kebutuhan tanpa mempengaruhi layanan lain.

#### **Penggunaan Sehari-Hari:**

- Saat menggunakan aplikasi streaming seperti Netflix, arsitektur di balik layar memungkinkan fitur-fitur yang berbeda tetap berjalan secara paralel dan independen.

## 6. Aplikasi Kolaboratif

Aplikasi seperti Google Docs atau Microsoft Teams memungkinkan beberapa pengguna untuk bekerja bersama secara real-time, dengan pengolahan data yang tersebar di berbagai server dan lokasi.

### Contoh:

- **Google Docs:** Setiap perubahan yang dilakukan oleh pengguna langsung disinkronkan di server Google, memungkinkan semua pengguna melihat update tersebut secara real-time.
- **Slack, Microsoft Teams:** Aplikasi ini mengintegrasikan komunikasi dan kolaborasi dengan banyak server yang memastikan ketersediaan dan sinkronisasi data antar pengguna di seluruh dunia.

### Kelebihan:

- **Real-time Collaboration:** Pengguna dapat berkolaborasi secara langsung tanpa lag.
- **Penyimpanan Terdistribusi:** Data disimpan secara otomatis di server, mengurangi risiko kehilangan data.

### Penggunaan Sehari-Hari:

- Bekerja bersama tim di dokumen atau spreadsheet Google.
- Mengadakan rapat atau kolaborasi tim melalui aplikasi seperti Microsoft Teams atau Slack.

## Kesimpulan

Sistem terdistribusi telah menjadi tulang punggung dari banyak teknologi modern yang kita gunakan setiap hari. Mulai dari layanan cloud hingga blockchain dan aplikasi P2P, sistem ini memungkinkan layanan yang lebih cepat, lebih andal, dan lebih terukur. Manfaat dari sistem terdistribusi dalam kehidupan sehari-hari mencakup kecepatan akses, keandalan, dan kemampuan untuk berkembang seiring meningkatnya permintaan.

### **C. POINT 3**

Arsitektur sistem terdistribusi adalah rangkaian komponen yang bekerja secara terpisah tetapi saling terhubung untuk mencapai suatu tujuan bersama. Komponen ini biasanya terdiri dari node atau server yang berkomunikasi melalui jaringan menggunakan protokol dan mekanisme komunikasi tertentu. Setiap implementasi sistem terdistribusi memiliki arsitektur spesifik yang disesuaikan dengan tujuan dan kebutuhan. Berikut adalah penjelasan tentang arsitektur umum yang digunakan dalam sistem terdistribusi, termasuk **komunikasi antar node**, **protokol yang digunakan**, dan **komponen penting lainnya**.

#### **1. Arsitektur Komunikasi Sistem Terdistribusi**

##### **a. Model Client-Server**

Model ini adalah salah satu arsitektur dasar dalam sistem terdistribusi, di mana server menyediakan layanan, dan klien mengakses layanan tersebut. Komunikasi terjadi melalui protokol jaringan seperti **HTTP**, **TCP/IP**, atau **WebSocket**.

- **Client:** Mengirim permintaan (request) ke server dan menunggu tanggapan (response).
- **Server:** Menerima permintaan dari klien, memprosesnya, dan mengirimkan hasil kembali ke klien.

##### **Contoh:**

- **Cloud Computing** seperti Google Drive atau Dropbox, di mana pengguna (klien) mengakses file dari server terdistribusi yang dikelola oleh penyedia layanan cloud.

##### **b. Model Peer-to-Peer (P2P)**

Dalam arsitektur P2P, semua node dalam jaringan memiliki peran yang sama, dan mereka dapat bertindak sebagai klien atau server. Node berkomunikasi langsung tanpa memerlukan server pusat.

- **Komunikasi:** Node saling berbagi data secara langsung, sering kali menggunakan protokol P2P seperti **BitTorrent** atau **Kademlia**.
- **Keuntungan:** Tidak ada satu titik pusat, sehingga lebih tahan terhadap kegagalan.

### Contoh:

- **BitTorrent**, di mana setiap pengguna yang memiliki file sebagian dapat menjadi "peer" dan membagikan file tersebut kepada pengguna lain tanpa server pusat.

### c. Model Publish-Subscribe

Pada model ini, komponen (publisher) mengirimkan pesan atau data ke "broker" atau "message queue", yang kemudian mendistribusikan pesan ke komponen lain (subscriber) yang berlangganan pesan tersebut. Model ini populer dalam aplikasi terdistribusi real-time.

- **Protokol:** Protokol populer yang digunakan adalah **MQTT** atau **Kafka** untuk mendukung messaging yang ringan dan skalabel.
- **Keuntungan:** Sangat efektif dalam sistem yang membutuhkan komunikasi real-time dengan banyak node.

### Contoh:

- **Facebook Notification System**, di mana setiap kali pengguna mengubah status atau menyukai kiriman, pemberitahuan dikirimkan ke semua subscriber secara real-time.

## 2. Protokol Komunikasi dalam Sistem Terdistribusi

Protokol komunikasi adalah aturan atau standar yang memungkinkan node dalam sistem terdistribusi untuk bertukar informasi dengan cara yang efisien. Beberapa protokol umum yang digunakan dalam sistem terdistribusi adalah:

### A. HTTP/HTTPS

Protokol yang digunakan oleh sebagian besar aplikasi web berbasis cloud. **HTTP** adalah protokol berbasis teks yang memungkinkan klien dan server untuk berkomunikasi. Versi **HTTPS** menambahkan lapisan keamanan melalui enkripsi.

- **Penggunaan:** Digunakan dalam layanan cloud computing, CDN, dan aplikasi berbasis web.

## B. TCP/IP

Protokol dasar dalam jaringan internet, **Transmission Control Protocol (TCP)** bertanggung jawab untuk memastikan pengiriman paket data yang andal, sementara **Internet Protocol (IP)** menangani pengalamatan dan routing paket data.

- **Penggunaan:** TCP/IP digunakan di hampir semua sistem terdistribusi untuk komunikasi antara node.

## C. gRPC (Google Remote Procedure Call)

**gRPC** adalah framework open-source yang digunakan untuk komunikasi antar layanan di arsitektur microservices. gRPC memungkinkan satu layanan untuk memanggil fungsi atau metode dari layanan lain yang berjalan di server yang berbeda.

- **Keuntungan:** Mendukung protokol HTTP/2, yang lebih efisien dalam hal latensi dan throughput.
- **Penggunaan:** Banyak digunakan dalam sistem berbasis microservices dan komunikasi antar layanan di cloud.

## D. Message Queues (AMQP, Kafka, MQTT)

Dalam arsitektur berbasis **publish-subscribe**, protokol seperti **AMQP (Advanced Message Queuing Protocol)**, **Apache Kafka**, dan **MQTT** sering digunakan untuk mendistribusikan pesan antar node secara terdistribusi dan terkelola dengan baik.

- **Keuntungan:** Protokol ini mendukung pengiriman pesan yang asinkron dan sangat skalabel.
- **Penggunaan:** Digunakan dalam aplikasi yang memerlukan komunikasi real-time, seperti notifikasi aplikasi, sensor IoT, dan microservices.

## E. Protokol Peer-to-Peer (P2P)

Protokol seperti **BitTorrent** atau **Kademlia** digunakan dalam sistem P2P untuk memungkinkan node berbagi data secara langsung tanpa server pusat. Protokol ini mengelola jaringan peer dan mendistribusikan data antar node secara efisien.

- **Keuntungan:** Desentralisasi dan efisiensi distribusi file.
- **Penggunaan:** Digunakan dalam jaringan berbagi file P2P seperti BitTorrent dan dalam beberapa implementasi blockchain.

### 3. Arsitektur Penyimpanan dalam Sistem Terdistribusi

#### A. Penyimpanan Terdistribusi (Distributed File System)

Dalam sistem terdistribusi, data sering kali disimpan di berbagai server yang tersebar di lokasi geografis yang berbeda. **Distributed File System (DFS)** adalah sistem yang memungkinkan akses ke file atau data dari berbagai lokasi dengan seolah-olah data berada di satu tempat.

- **Contoh: Google File System (GFS)** dan **Hadoop Distributed File System (HDFS)** adalah contoh DFS yang digunakan dalam cloud dan big data analytics.

#### B. Database Terdistribusi (Distributed Database)

Database terdistribusi adalah basis data yang disimpan pada beberapa node, di mana setiap node bertanggung jawab atas sebagian data. Komunikasi antar node menjaga konsistensi dan integritas data.

- **Contoh: Cassandra, MongoDB, dan Amazon DynamoDB** adalah contoh database terdistribusi yang digunakan dalam aplikasi berskala besar dan real-time.

#### C. Blockchain

Blockchain adalah teknologi penyimpanan terdistribusi di mana setiap node dalam jaringan memelihara salinan lengkap dari ledger atau buku besar yang berisi transaksi yang telah diverifikasi.

- **Protokol: Proof of Work (PoW)** atau **Proof of Stake (PoS)** adalah mekanisme yang digunakan untuk memverifikasi transaksi di jaringan blockchain.
- **Keuntungan:** Keamanan tinggi dan desentralisasi penuh, tetapi sering kali lambat dalam memproses transaksi dibandingkan dengan sistem terpusat.

## 4. Komponen Kunci dalam Sistem Terdistribusi

### A. Load Balancer

Dalam sistem terdistribusi, **load balancer** digunakan untuk mendistribusikan beban kerja secara merata di antara server yang berbeda, sehingga mencegah satu server dari menjadi terlalu sibuk sementara yang lain menganggur.

- **Contoh:** Load balancer yang umum digunakan termasuk **HAProxy** dan **Nginx**.

### B. Service Discovery

Sistem ini memungkinkan layanan di jaringan untuk menemukan layanan lain tanpa perlu mengetahui lokasi fisik mereka. Service discovery sering digunakan dalam arsitektur microservices.

- **Contoh:** **Consul** dan **Etcd** digunakan untuk service discovery dalam sistem terdistribusi.

### C. Failover dan Replication

Untuk menjaga ketersediaan tinggi, sistem terdistribusi sering kali menggunakan teknik **failover** (peralihan otomatis ke cadangan saat server utama gagal) dan **replication** (menyimpan salinan data di beberapa lokasi).

- **Contoh:** Sistem seperti **Kafka** mendukung replikasi data untuk memastikan bahwa data tetap tersedia bahkan jika salah satu node gagal.

## Kesimpulan

Arsitektur sistem terdistribusi terdiri dari banyak komponen yang berkomunikasi melalui jaringan, menggunakan berbagai protokol seperti HTTP, gRPC, MQTT, dan lainnya. Protokol-protokol ini memungkinkan sistem untuk tetap efisien, skalabel, dan reliabel. Dengan komponen seperti load balancer, service discovery, dan failover, sistem ini dapat mengatasi tantangan teknis terkait distribusi data dan beban kerja, serta menjaga ketersediaan layanan di seluruh node.

Protokol, teknik penyimpanan, dan model komunikasi yang tepat sangat bergantung pada kebutuhan aplikasi dan jenis sistem yang diimplementasikan.

## DAFTAR PUSTAKA

**Tanenbaum, A. S., & Van Steen, M. (2017).** *Distributed Systems: Principles and Paradigms*. Prentice Hall.

**Silberschatz, A., Galvin, P. B., & Gagne, G. (2018).** *Operating System Concepts* (10th ed.). Wiley.

**Coulouris, G., Dollimore, J., & Kindberg, T. (2011).** *Distributed Systems: Concepts and Design* (5th ed.). Addison-Wesley.

**Mell, P., & Grance, T. (2011).** *The NIST Definition of Cloud Computing*. National Institute of Standards and Technology.

**Harrison, G., & Fritchie, B. (2020).** *Content Delivery Networks (CDN): An Introduction*. Web Performance, Inc.

**Fowler, M. (2015).** *Microservices: A Definition of this New Architectural Term*. martinofowler.com.

**Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010).** *The Hadoop Distributed File System*. 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST).

**Gotsman, A., & M. V. (2013).** *The Ethereum Blockchain: A Distributed Platform for Secure Applications*. ACM Transactions on Computational Logic.