

# Sistem Terdisreibusi



## Komunikasi Jarak Jauh

TKB6252 – Sistem Terdistribusi

**Chalifa Chazar**

**[www.script.id](http://www.script.id)**

**[chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)**

# Pendahuluan

- Komunikasi antar komputer dari vendor yang berbeda sangat sulit dilakukan
- Karena masing-masing menggunakan **protokol** dan format data yang berbeda-beda
- Hingga ISO (International Standards Organization) membuat suatu bentuk arsitektur komunikasi yang disebut **OSI (Open System Interconnection)**
- **OSI** menyebabkan hubungan antar komputer-komputer dari vendor-vendor yang berbeda dapat dilakukan

# Protokol

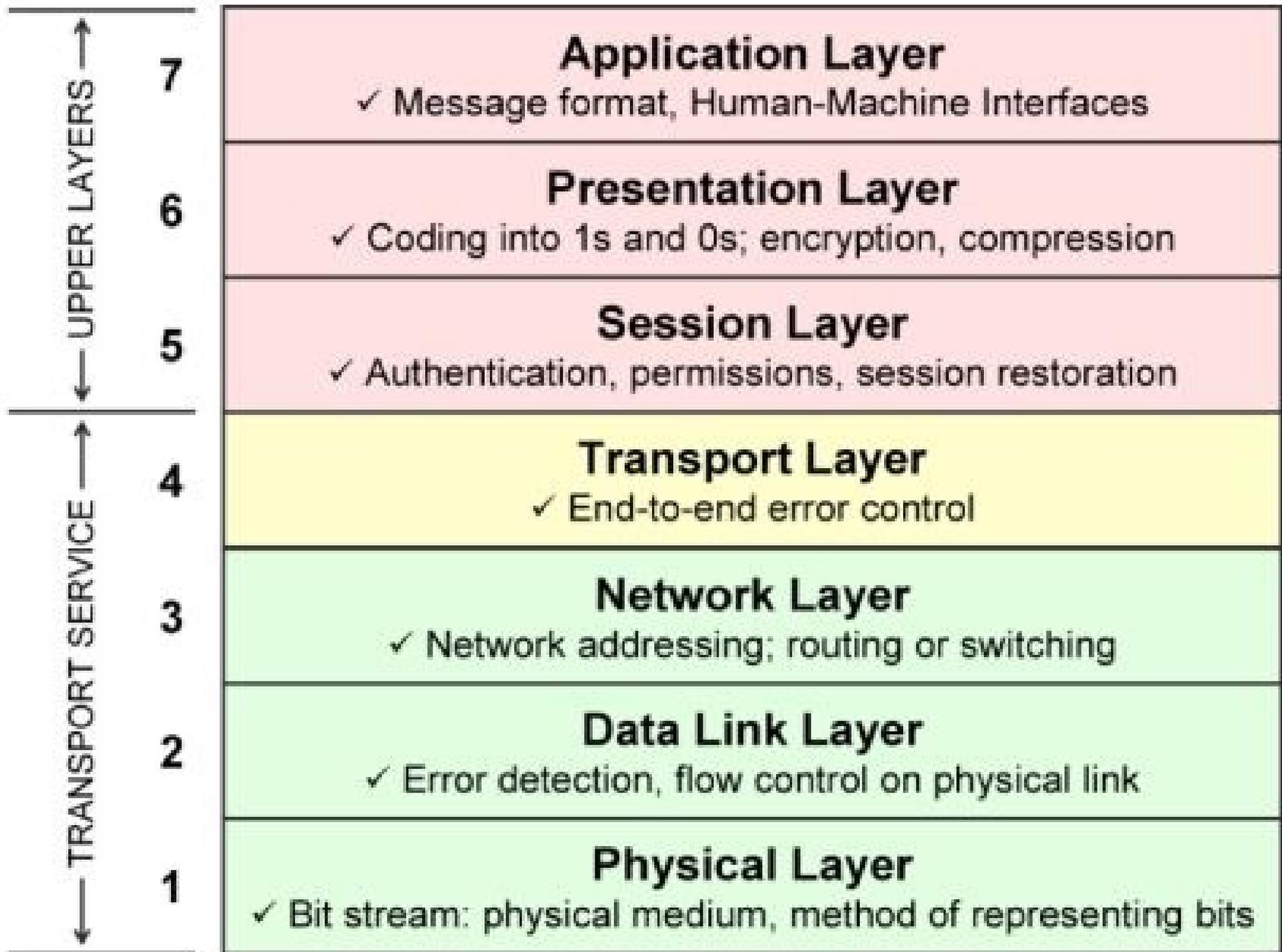
- Protokol adalah sebuah aturan standar yang mengatur/mengijinkan terjadinya hubungan, komunikasi dan perpindahan data antara dua atau lebih titik komputer
- Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya

# OSI

- OSI adalah model yang mendefinisikan **standar untuk komunikasi yang menyebabkan seluruh alat komunikasi dapat saling berkomunikasi melalui jaringan** secara efisien
- OSI layer menggambarkan bagaimana informasi dari suatu aplikasi di sebuah komputer berpindah melewati sebuah media jaringan ke suatu aplikasi di komputer lain
- Terbagi dalam 2 kelompok yaitu:
  - **Upper layer**, fokus pada aplikasi pengguna dan bagaimana file direpresentasikan di komputer
  - **Lower layer (transport service)**, intisari komunikasi data melalui jaringan aktual

# Tujuan OSI Layer

Tujuan utama OSI Layer adalah **untuk membantu desainer jaringan memahami fungsi dari tiap-tiap layer yang berhubungan dengan aliran komunikasi data** (termasuk jenis-jenis protokol jaringan dan metode transmisi)



## OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example	Central Device/ Protocols	DOD4 Model
<b>Application (7)</b> Serves as the window for users and application processes to access the network services.	<b>End User layer</b> Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management	<b>User Applications</b>  SMTP	<b>G A T E W A Y</b>
<b>Presentation (6)</b> Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	<b>Syntax layer</b> encrypt & decrypt (if needed)  Character code translation • Data conversion • Data compression • Data encryption • <b>Character Set Translation</b>	JPEG/ASCII EBDIC/TIFF/GIF PICT	
<b>Session (5)</b> Allows session establishment between processes running on different stations.	<b>Synch &amp; send to ports</b> (logical ports)  Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.	<b>Logical Ports</b>  RPC/SQL/NFS NetBIOS names	
<b>Transport (4)</b> Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	<b>TCP</b> Host to Host, Flow Control  Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	<b>F I L T E R I N G</b>	TCP/SPX/UDP  Host to Host
<b>Network (3)</b> Controls the operations of the subnet, deciding which physical path the data takes.	<b>Packets</b> ("letter", contains IP address)  Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		<b>Routers</b>  IP/IPX/ICMP  Internet
<b>Data Link (2)</b> Provides error-free transfer of data frames from one node to another over the Physical layer.	<b>Frames</b> ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control	<b>Switch Bridge WAP</b> PPP/SLIP	Can be used on all layers
<b>Physical (1)</b> Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	<b>Physical structure</b> Cables, hubs, etc.  Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts	<b>Hub</b>  Land Based Layers	

# Model OSI

- Layer tersebut disusun sedemikian rupa sehingga perubahan pada satu layer tidak membutuhkan perubahan pada layer lain.
- Layer teratas (5, 6, dan 7) merupakan layer yang lebih cerdas, application layer dapat menangani protokol dan format data yang sama yang digunakan oleh layer lain

# Physical Layer

- Merupakan layer sederhana yang berkaitan dengan electrical (dan optical) koneksi antar peralatan
- Physical layer ini bertanggung jawab untuk mentransmisikan bit data digital dari physical layer perangkat pengirim (sumber) menuju ke physical layer perangkat penerima (tujuan) melalui media komunikasi jaringan.
- Data ditransmisikan menggunakan jenis sinyal yang didukung oleh media fisik, seperti tegangan listrik, kabel, frekuensi radio atau infrared maupun cahaya biasa.

# Data Link Layer

- Data Link Layer bertanggung jawab untuk memeriksa kesalahan yang mungkin terjadi pada saat proses transmisi data dan juga membungkus bit kedalam bentuk data frame
- Data Link layer juga mengelola skema pengalamatan fisik seperti alamat MAC pada suatu jaringan
- Merupakan salah satu layer OSI yang cukup kompleks, oleh karena itu layer ini kemudian dibagi lagi menjadi dua sub-layer, yaitu:
  - **Layer Media Access Control (MAC)** = bertanggung jawab untuk mengendalikan bagaimana sebuah perangkat pada suatu jaringan memperoleh akses ke medium dan izin untuk melakukan transmisi data
  - **Layer Logical Link Control (LLC)** = bertanggung jawab untuk mengidentifikasi dan membungkus protokol network layer dan mengontrol pemeriksaan kesalahan dan juga melakukan sinkronisasi pada frame

# Network Layer

- Network layer bertanggung jawab untuk menetapkan jalur yang akan digunakan untuk melakukan transfer data antar perangkat didalam suatu jaringan
- Router jaringan beroperasi pada layer ini, yang mana juga menjadi fungsi utama pada layer network dalam hal melakukan routing.
- Routing memungkinkan paket dipindahkan antar komputer yang terhubung satu sama lain
- Untuk mendukung proses routing ini, network layer menyimpan alamat logis seperti alamat IP untuk setiap perangkat pada jaringan
- Layer Network juga mengelola pemetaan antara alamat logikal dan alamat fisik
- Dalam jaringan IP, pemetaan ini dilakukan melalui Address Resolution Protocol (ARP).

# Transport Layer

- Transport layer bertanggung jawab untuk mengirimkan pesan antara dua atau lebih host didalam jaringan
- Transport layer juga menangani pemecahan dan penggabungan pesan dan juga mengontrol kehandalan jalur koneksi yang diberikan
- Protokol TCP merupakan contoh yang paling sering digunakan pada transport layer.

# Session Layer

- Session layer bertanggung jawab untuk mengendalikan sesi koneksi dialog seperti menetapkan, mengelola dan memutuskan koneksi antar komputer
- Untuk dapat membentuk sebuah sesi komunikasi, session layer menggunakan sirkuit virtual yang dibuat oleh transport layer

# Presentation Layer

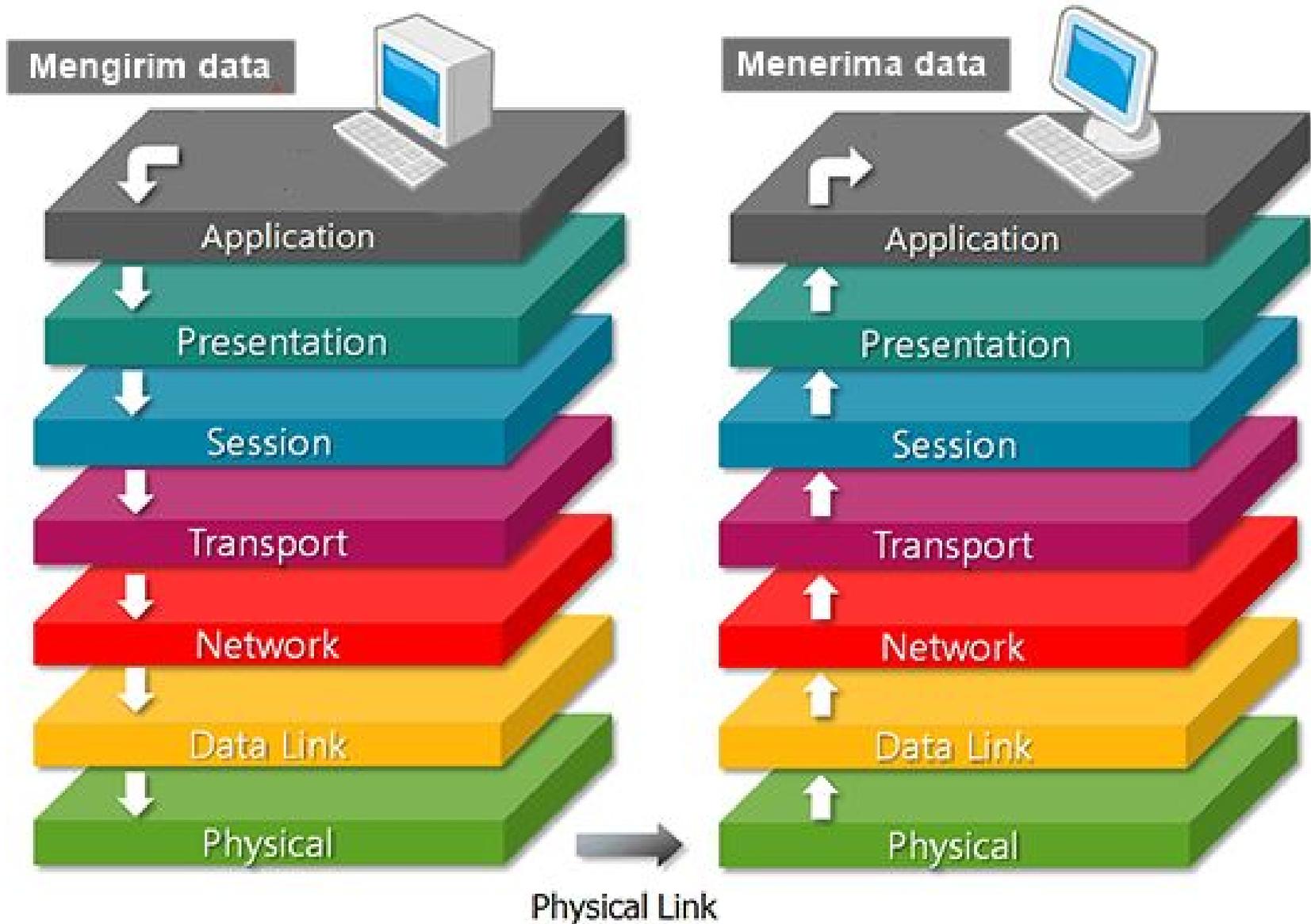
- Presentation layer bertanggung jawab untuk mendefinisikan sintaks yang digunakan host jaringan untuk berkomunikasi
- Presentation layer juga melakukan proses enkripsi/dekripsi informasi atau data sehingga mampu digunakan pada lapisan aplikasi

# Application Layer

- Application layer merupakan lapisan paling atas dari model OSI dan bertanggung jawab untuk menyediakan sebuah interface antara protokol jaringan dengan aplikasi yang ada pada komputer
- Application layer menyediakan layanan yang dibutuhkan oleh aplikasi, seperti menyediakan sebuah interface untuk Simple Mail Transfer Protocol (SMTP), telnet dan File Transfer Protocol (FTP)
- Pada bagian inilah aplikasi saling terkait dengan jaringan.

<b>LAYER</b>	<b>FUNGSI</b>	<b>CONTOH</b>
<b>Application (7)</b>	Menunjang aplikasi untuk berkomunikasi melalui jaringan	SMTP
<b>Presentation (6)</b>	Memformat data sehingga dapat dikenali oleh penerima	JPG, GIF, HTTPS, SSL, TLS
<b>Session (5)</b>	Membentuk koneksi, kemudian memutuskannya ketika seluruh data telah terkirim	NetBIOS, PPTP
<b>Transport (4)</b>	Mengatur flow control, acknowledgment dan mengirim ulang data jika diperlukan	TCP, UDP
<b>Network (3)</b>	Menambahkan alamat jaringan pada paket	Router, Layer 3 Switch
<b>Data Link (2)</b>	Menambahkan MAC address pada paket	Switch
<b>Physical (1)</b>	Mengirimkan data melalui media transmisi	Hub, NIC, Kabel

# CARA KERJA OSI LAYER



# Masih Ingat?

- Salah satu contoh tugas dari sistem operasi terdistribusi adalah menyediakan lingkungan untuk login jarak jauh
- Bagaimana suatu client dapat mengakses layanan yang ada di server
- Proses layanan jauh ini menggunakan suatu metode RPC (Remote Procedure Call)

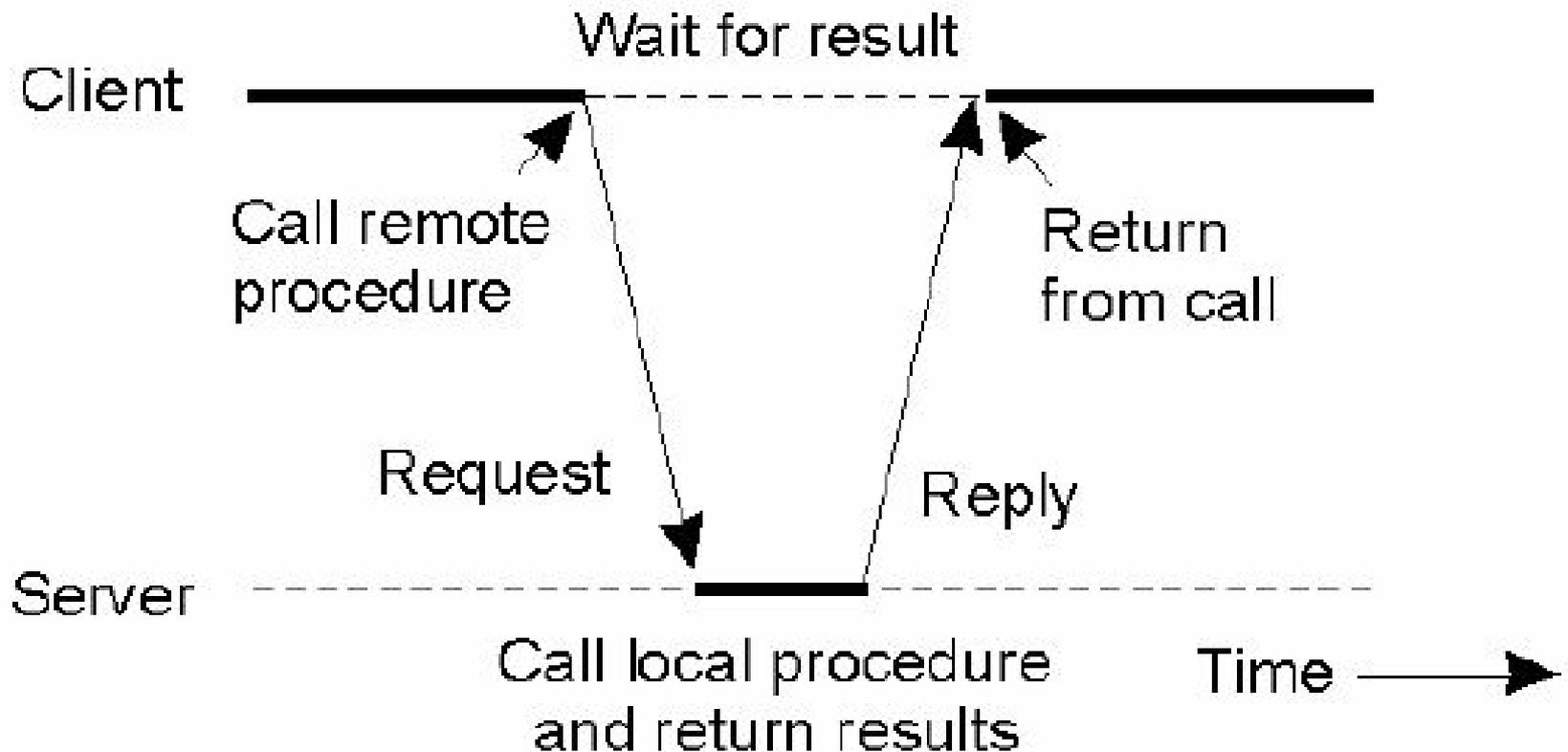
# RPC

- Remote Procedure Call (RPC) adalah sebuah metode yang memungkinkan kita untuk mengakses sebuah prosedur yang berada pada komputer lain
- Untuk dapat melakukan ini, komputer (server) harus menyediakan layanan remote procedure
- Pendekatan yang dilakukan adalah, sebuah server membuka socket, menunggu client yang meminta prosedur yang disediakan oleh server

# RPC (2)

- RPC masih menggunakan cara primitive dalam pemograman, yaitu menggunakan paradigma procedural programming
- Hal itu menyebabkan kita sulit ketika harus menyediakan banyak remote
- RPC menggunakan socket untuk berkomunikasi dengan proses lainnya
- Umumnya protocol RPC yang digunakan saat ini adalah DCOM (Distributed Component Object Model)
- Alternatif protrocol lainnya adalah SOAP (Simple Object Access Protocol) yang menggunakan teknologi XML

# Prinsip RPC Dalam Program Clien-Server



# Kekurangan RPC

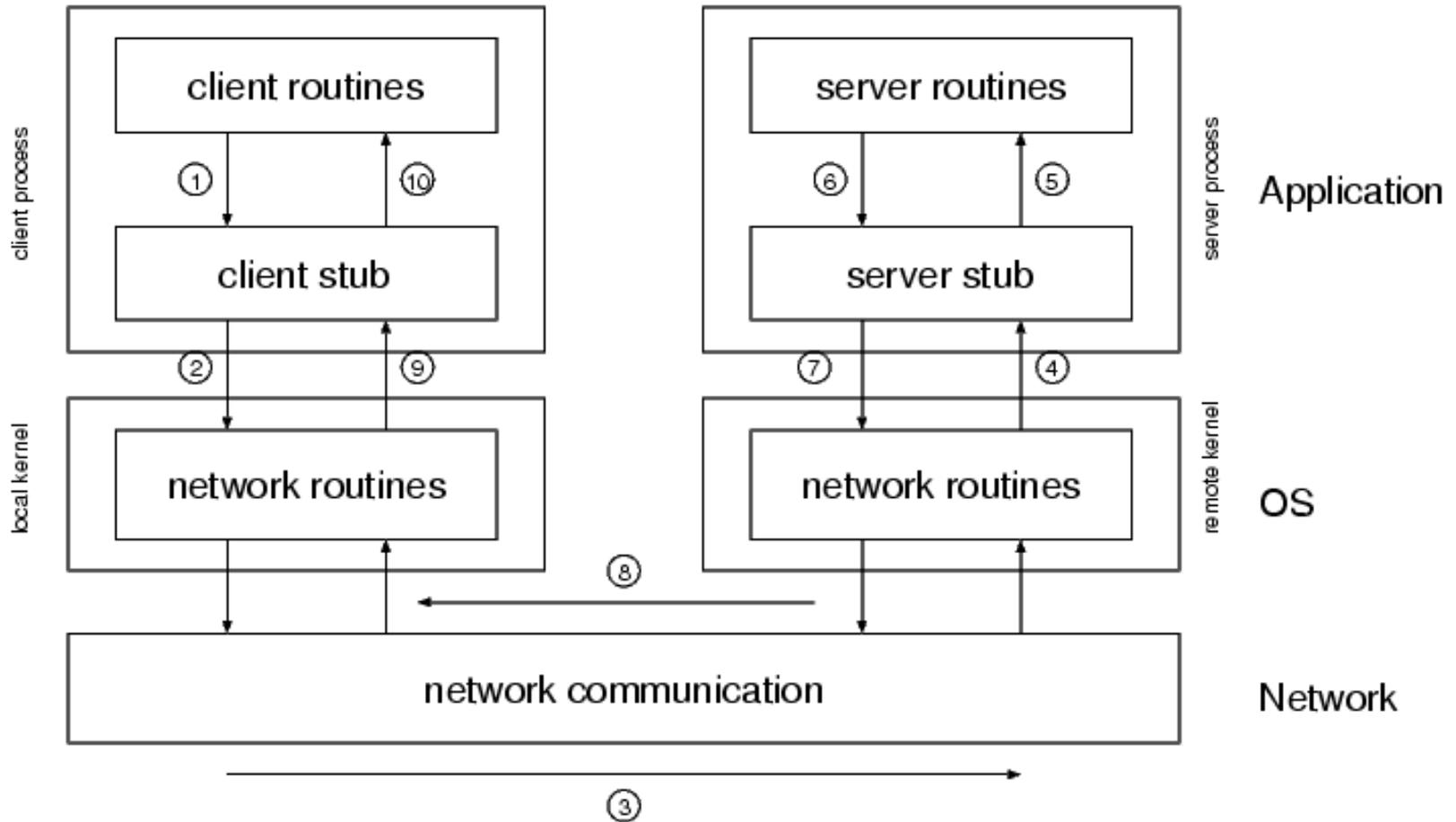
- **Tidak fleksibel terhadap perubahan**
  - Static relationship between client & server at run-time
  - Berdasarkan prosedural/structured programming yang sudah ketinggalan jaman dibandingkan OOP.
- **Kurangnya location transparency**
  - Misalnya pemrograman hanya boleh melakukan pass by value, bukan pass by reference
  - Komunikasi hanya antara 1 klien & 1 server (one-to-one at a time)
  - Komunikasi antara 1 klien & beberapa server memerlukan beberapa koneksi yang terpisah

# Kelebihan RPC

- **Relatif mudah digunakan**
  - Pemanggilan remote procedure tidak jauh berbeda dibandingkan pemanggilan local procedure
  - Sehingga pemrogram dapat berkonsentrasi pada software logic, tidak perlu memikirkan low level details seperti soket, marshalling & unmarshalling.
- **Robust (sempurna)**
  - RPC telah banyak digunakan dlm pengembangan mission-critical application yang memerlukan scalability, fault tolerance & reliability.

# Langkah-Langkah Dalam RPC

## Remote Procedure Calls



1.	Client memanggil local procedure, yang disebut client stub. Bagi client, client stub seolah-olah server procedure yang dipanggil. Stub melakukan marshalling (argumen untuk remote procedure diatur dalam paket dengan mengikuti suatu format standar, menjadi sebuah network message)
2.	Melalui pemanggilan sistem call di local kernel, client stub mengirim network message ke remote sistem
3.	Network message ditransfer ke remote sistem, menggunakan connection-oriented maupun connectionless protocol
4.	Server stub menunggu request dari client, melakukan unmarshalling argumen yang ada di network message, dan melakukan konversi jika diperlukan
5.	Server stub menjalankan local procedure call yang memanggil server procedure yang sebenarnya
6.	Setelah server procedure selesai, return value dikembalikan ke server stub
7.	Server stub melakukan konversi (jika diperlukan), kemudian melakukan marshalling menjadi network message, dan mengirimkannya kembali ke client stub
8.	Network message ditransfer kembali ke client stub
9.	Client stub membaca network message dari local kernel
10	Client stub mengkonversi return value (jika diperlukan), kemudian mengembalikannya ke client routine, seperti return value dari local procedure

# RMI

- Remote Method Invocation (RMI), sebuah teknik pemanggilan metode request yang secara umum lebih baik dari RPC
- RMI menggunakan paradigma pemrograman berorientasi objek (OOP)
- RMI memungkinkan kita untuk mengirim objek sebagai parameter dari remote method
- RMI memungkinkan pengguna dapat mengembangkan aplikasi java yang terdistribusi pada jaringan

# RMI (2)

- Untuk membuat remote method dapat diakses RMI mengimplementasikan remote object menggunakan stub dan skleton
- Stub bertindak sebagai proxy disisi client, yaitu yang menghubungkan client dengan skelon yang berada disisi server
- Untuk membuat remote obyek kita harus mendefinisikan semua method yang akan kita sediakan pada jaringan, setelah itu dapat digunakan RMI compiler untuk membuat stub dan skleton
- Setelah itu kita harus mem-binding remote obyek yang kita sediakan kedalam sebuah RMI registry
- Setelah itu client dapat mengakses semua remote method yang telah kita sediakan menggunakan stub yang telah di-compile menggunakan RMI compiler tersebut.
- RMI sendiri merupakan sistem terdistribusi yang dirancang oleh SUN pada platfrom yang spesifik yaitu Java

# Kesimpulan

- RPC dirancang sebagai suatu cara untuk mengabstraksikan mekanisme procedure-call untuk digunakan antar sistem dengan jaringan terhubung
- RPC merupakan suatu procedure yang letaknya di middleware layer (antara transport layer dengan application layer)
- RPC sebuah proses pada local sistem memanggil prosedur pada remote sistem, namun bagi pemogram, seperti melakukan local procedure call
- Tujuan digunakannya RPC adalah untuk memudahkan pengembangan aplikasi terdistribusi, client-server tidak perlu memikirkan network details, seperti socket



**</TERIMA KASIH>**

Chalifa Chazar, S.T, M.T

Email: [chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)

script.id

Copyright @2016