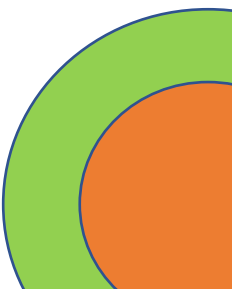


LAYER APLIKASI

PERTEMUAN KE 9

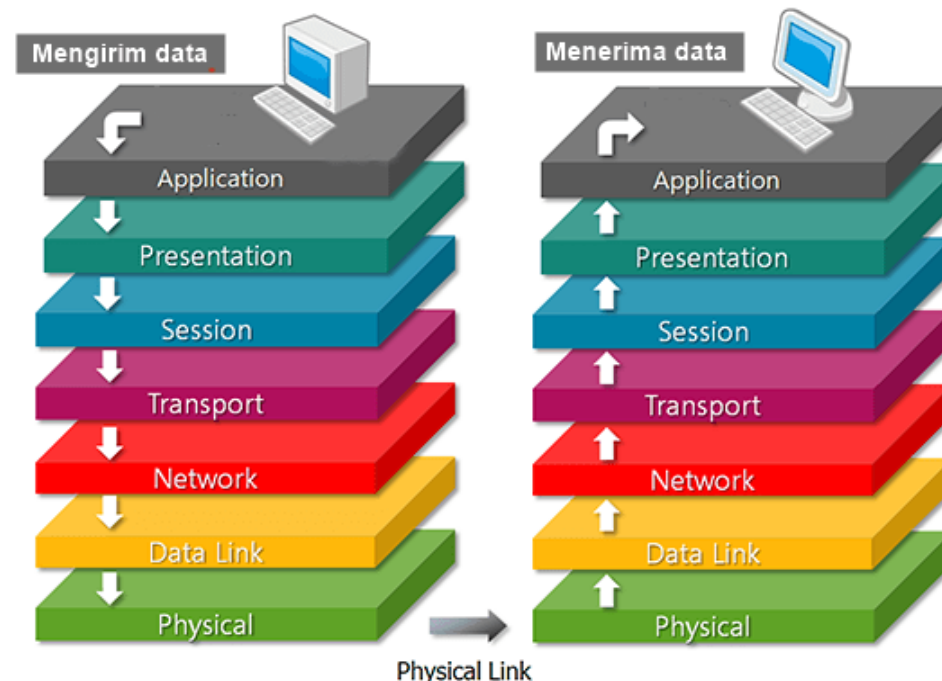
MATA KULIAH JARINGAN KOMPUTER



PERAN LAYER APLIKASI DI JARINGAN KOMPUTER

Application Layer berperan untuk menyediakan layanan ke pengguna, dimana komunikasi dilakukan dengan menggunakan *logical connection*.

CARA KERJA OSI LAYER



Koneksi logis adalah suatu koneksi yang menggunakan protokol atau aturan yang digunakan oleh komputer agar dapat berkomunikasi dengan komputer lainnya.

Gambar 9.1 Hubungan Client Server menggunakan Layer OSI

PARADIGMA JARINGAN KOMPUTER PADA LAYER APLIKASI



Kedua komputer yang terhubung dapat berada pada lokasi yang berjauhan. Di internet, dapat diumpamakan dua buah program yang bekerja pada dua buah komputer yang berbeda dan saling berinteraksi.



Komputer yang terhubung dapat saling mengirim pesan melalui infrastruktur internet

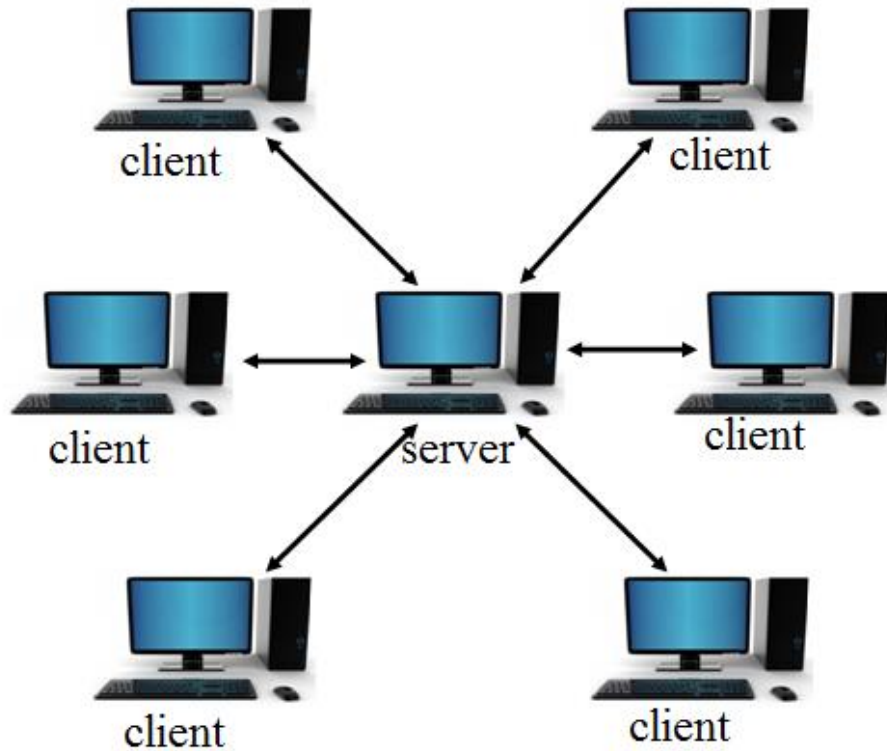
Gambar 9.2 Jaringan internet

Tiga Jenis Paradigma pada Layer Aplikasi

- **Client Server**
- **Peer to Peer**
- **Mixed (campuran)**



Paradigma Client Server



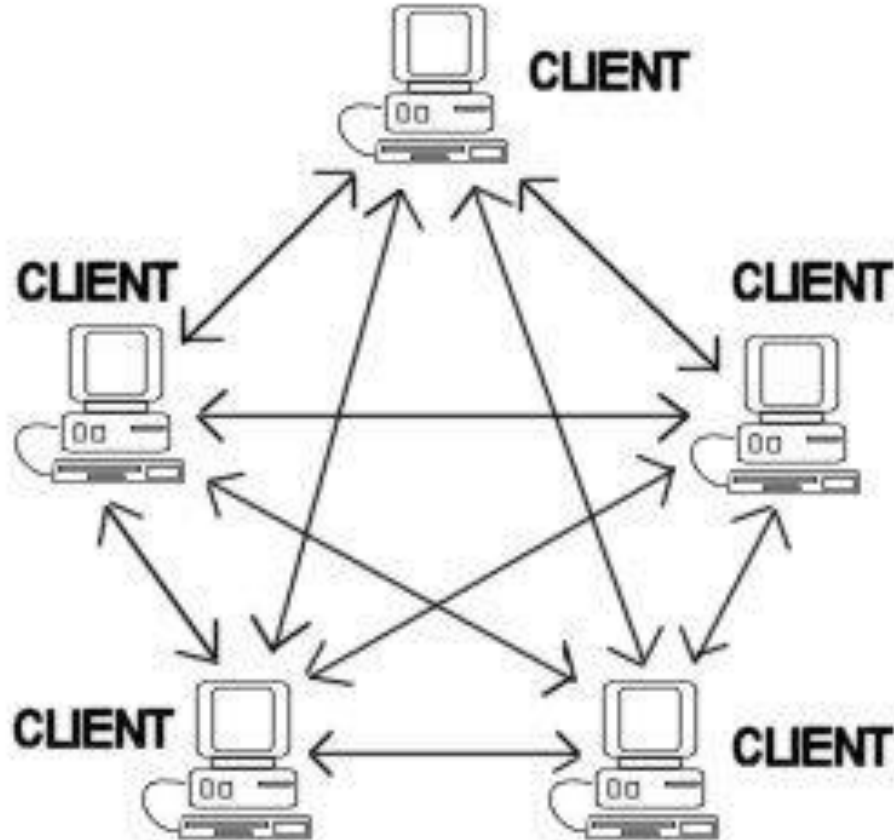
Gambar 9.3 Paradigma *Client Server*

Pada paradigma *client server* ini, dilakukan dua program yang sedang berjalan, baik disisi server maupun disisi *Client*. Masing-masing terdapat proses atau program/aplikasi yang berjalan

Server bertugas melayani permintaan *client*, sedangkan *client* bertugas mengirimkan permintaan pada server.



Paradigma Peer to Peer (P2P)

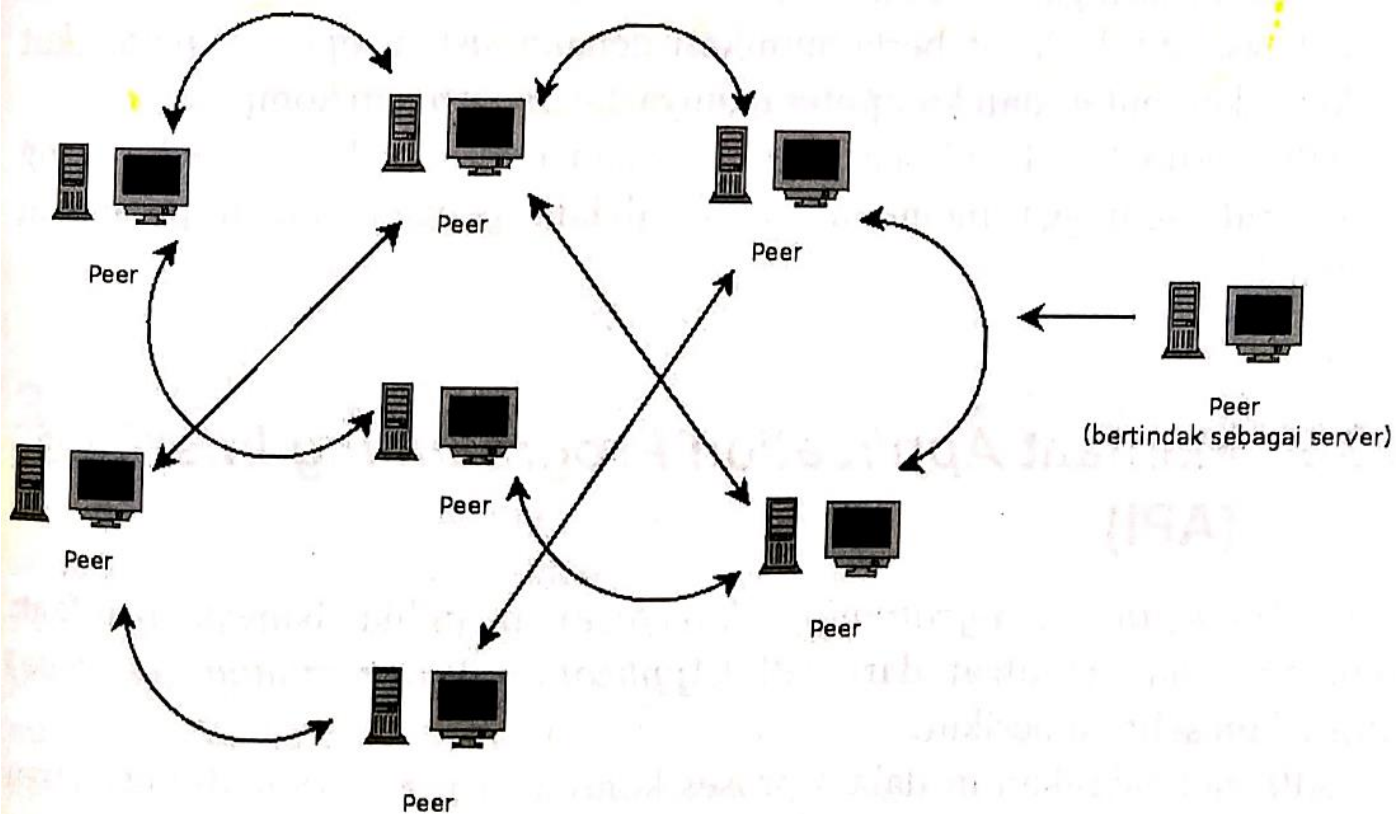


Paradikma ini tidak diperlukan adanya server untuk melayani permintaan client. Semua komputer yang berada didalam jaringan, dapat bertindak sebagai server sekaligus client.

Gambar 9.4 Paradigma *Peer to Peer*



Paradigma Campuran (Mixed)



Paradigma ini dilakukan kombinasi dari keuntungan-keuntungan yang ditawarkan oleh paradigma *client server* dan paradigma *peer to peer*.

Gambar 9.5 Paradigma *Mixed*

Sebagai contoh: pada *Light Load Client Server Communication* (LLCSC), dimana *server* bertugas menentukan *peer* mana yang mampu memberikan layanan ke *client*, untuk kemudian *peer (client)* bersangkutan yang akan memberikan langsung layanannya.



APPLICATION PROGRAMMING INTERFACE (API)



Agar proses di *client* dan proses di *server* dapat saling berkomunikasi dengan baik, maka perlu adanya sejumlah instruksi didalam program/aplikasi yang dijalankan.

Program yang ditulis dengan bahasa pemrograman, didalamnya berisi sekumpulan instruksi untuk operasi matematika, manipulasi string dan lain-lain. Sehubungan dengan bahasa pemrograman tentulah terdapat **API** (*Application Programming Interface*)

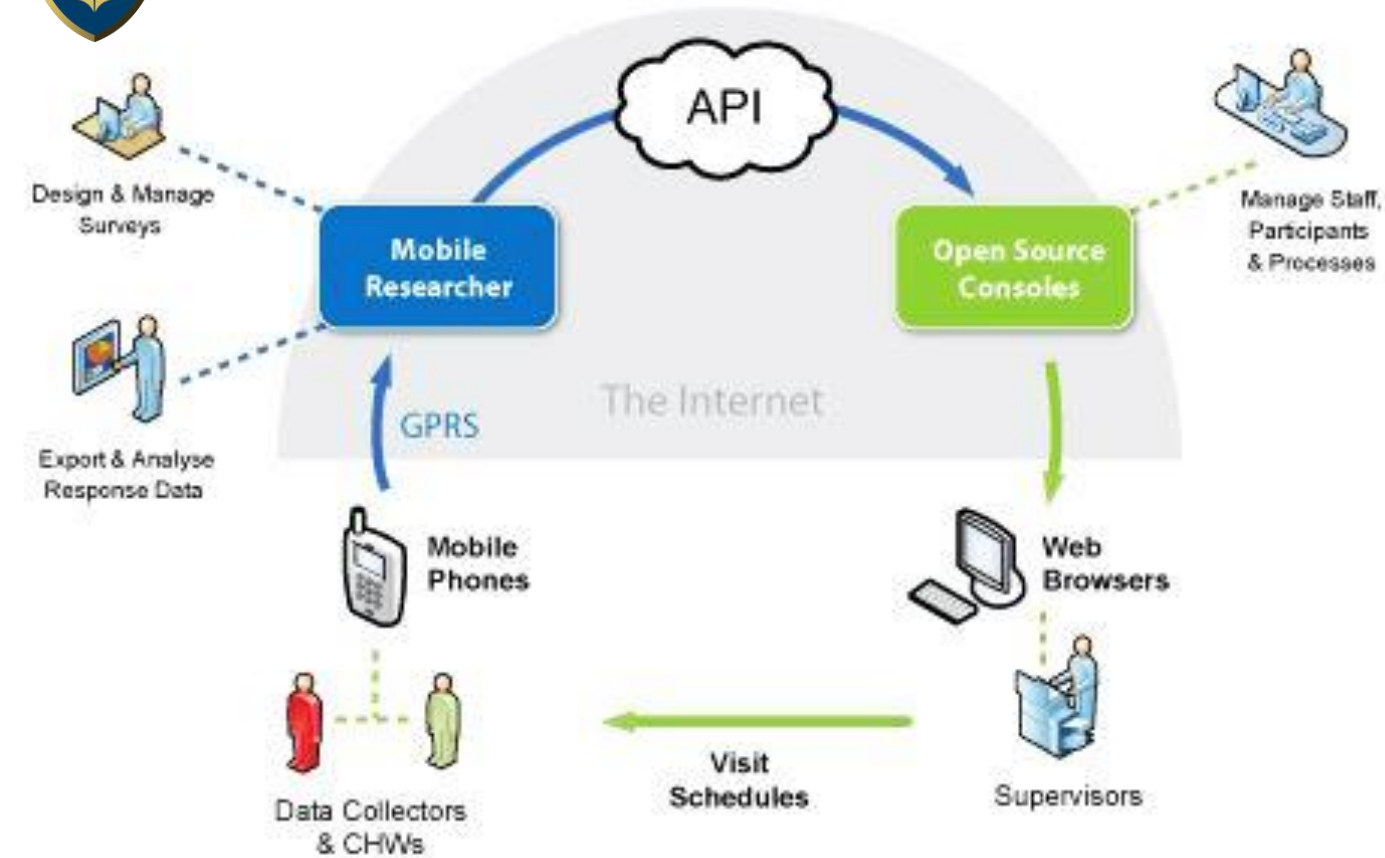


Definisi tatap muka program aplikasi atau API:



1. API merupakan sekumpulan instruksi yang berjalan di sisi layer aplikasi dan di sisi sistem operasi
2. API merupakan sekumpulan dari beragam perintah, fungsi dan protokol didalam jaringan komputer, agar aplikasi dapat berkomunikasi dengan sistem operasi, perangkat keras, dan komputer lain dalam suatu jaringan komputer
3. API merupakan kombinasi dari beragam perintah dan prosedur yang terurut, sehingga memudahkan dalam pengembangan perangkat lunak





Gambar 9.6 *Application Programming Interface*

Manfaat API :

1. Memudahkan didalam proses komunikasi, meliputi *socket interface*, *Transport Layer Interface* (TLI), dan *Stream*.
2. Memudahkan programmer dalam pembuatan aplikasi, melalui sejumlah prosedur, fungsi, tombol, pustaka yang disediakan dan digunakan
3. Membantu aplikasi untuk dapat berkomunikasi dengan sistem operasi, perangkat keras komputer, dan juga komputer-komputer lain yang terhubung dalam suatu jaringan komputer



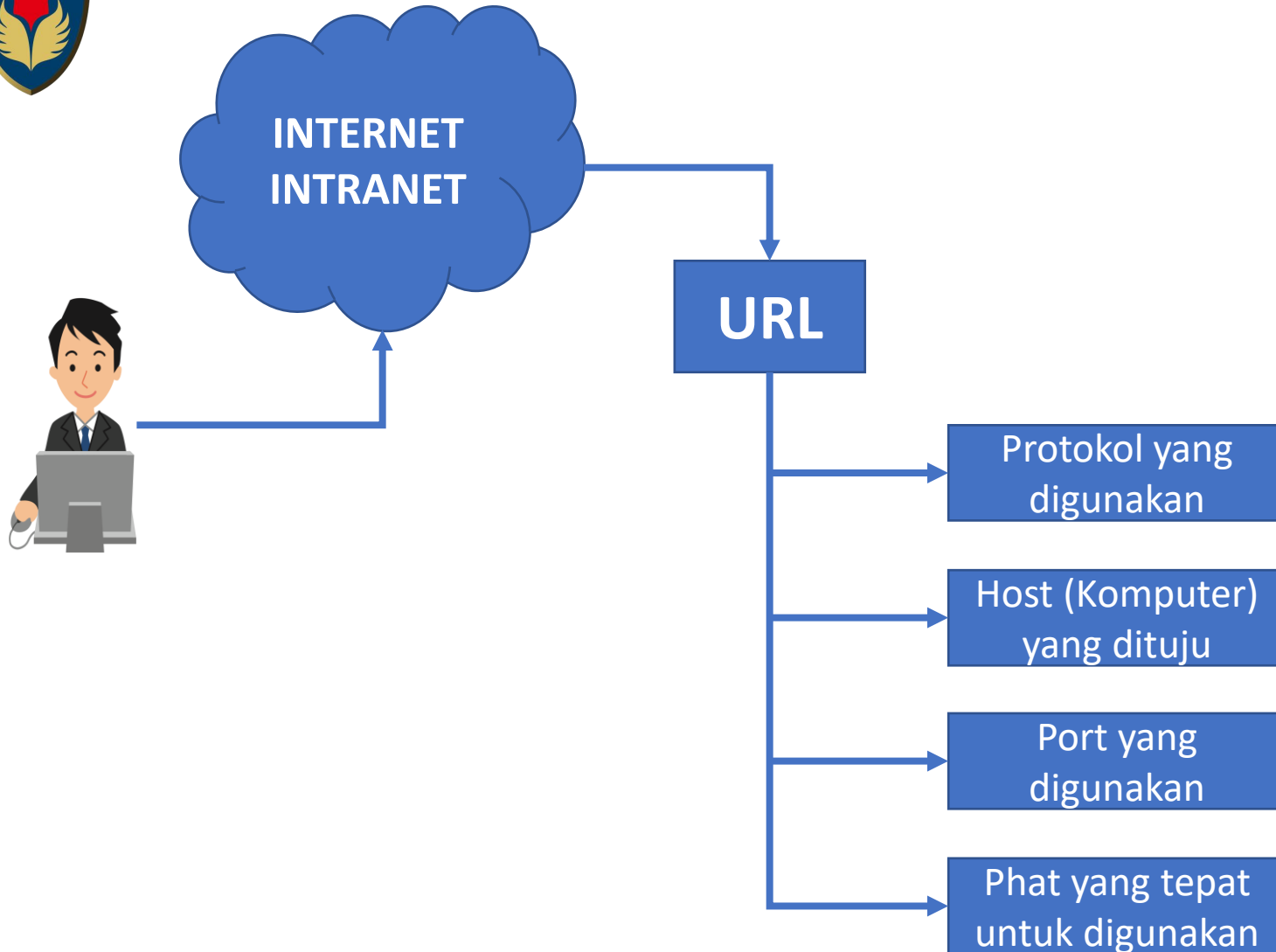
APLIKASI-APLIKASI STANDAR *CLIENT SERVER*

Seiring dengan perkembangan teknologi dan jaringan komputer, maka semakin banyak aplikasi-aplikasi berbasis *client server* yang dikembangkan. Beberapa buah aplikasi standar yang berbasis *client server* pada jaringan komputer, beberapa diantaranya juga menjadi sebuah protokol di dalam jaringan komputer.

WWW dan HTTP

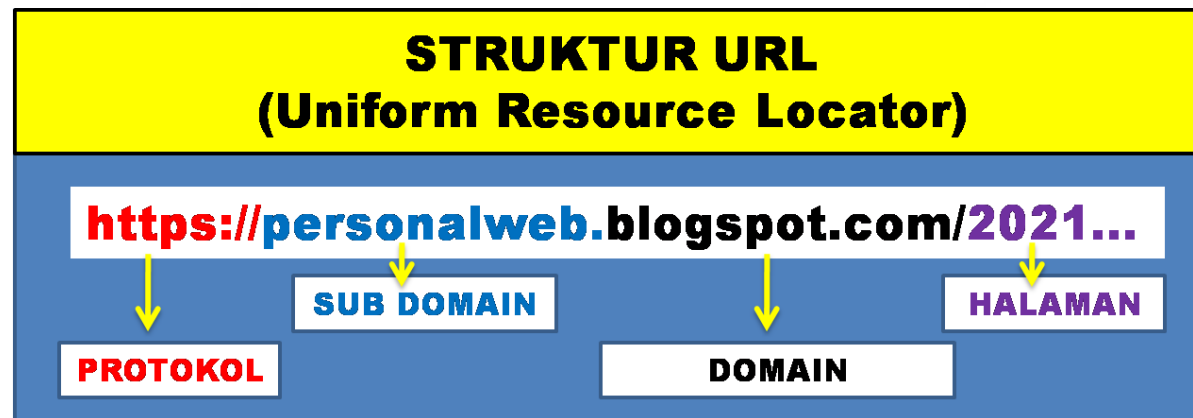
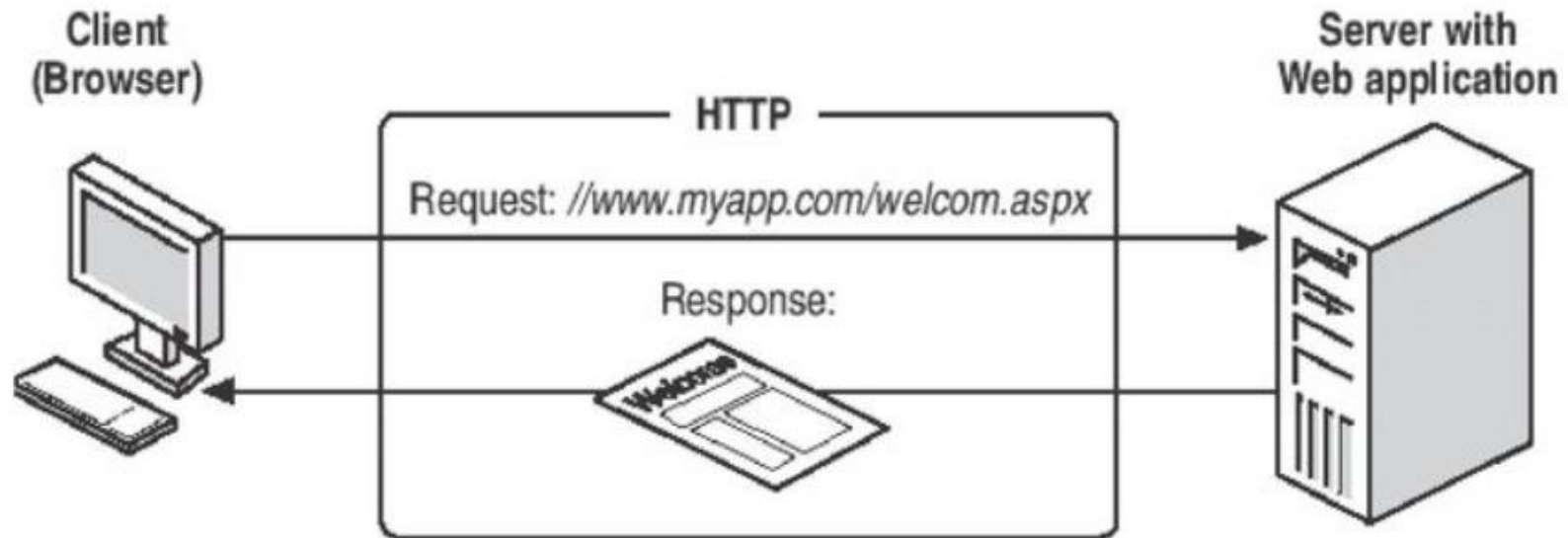
WWW merupakan contoh aplikasi berbasis *client – Server*, dimana *client* melakukan permintaan layanan kepada *server* melalui aplikasi *web browser*.

WWW (*Word Wide Web*) atau lebih dikenal sebagai *web* selalu berkaitan dengan HTTP (*Hyper Text Transfer Protocol*) dan terkadang HTTPS (*Hyper Text Transfer Protocol Secure*), sebagai protokol yang digunakannya.



Gambar 9.7 Bagan umum Uniform Resource Locator (URL)

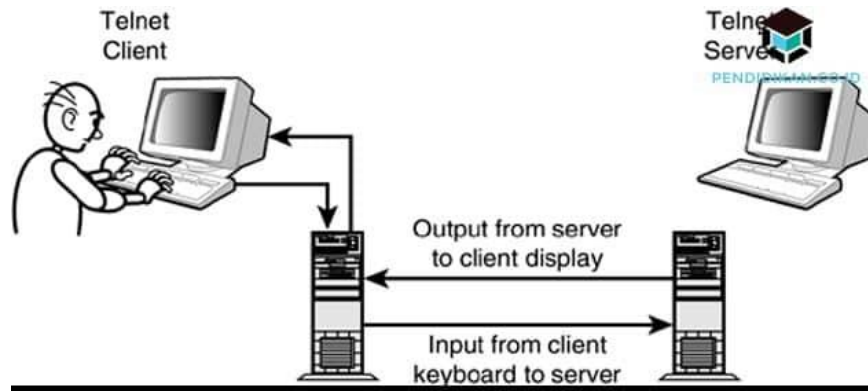
- Protokol yang digunakan harus jelas mencantumkan protokol apa yang digunakan di dalam melakukan koneksi HTTP, HTTPS atau FTP
- Hos (komputer) yang dituju, harus mencantumkan alamat dari komputer server pemberi layanan
- Port yang digunakan memberikan layanan penting di dalam koneksi pada jaringan komputer. *Server* akan membukakan port koneksi, sehingga semua klien dapat terhubung didalamnya
- Path yang tepat dari alamat tujuan untuk memperoleh data/file/layanan. Contoh :
<https://192.168.1.3:3378>
http artinya kita menggunakan protokol HTTPS, alamat IP 192.168.1.3 menyatakan alamat komputer tujuan (host) penyedia layanan, nomor 3378 menyatakan alamat spesifik yang digunakan oleh komputer server yang dituju.



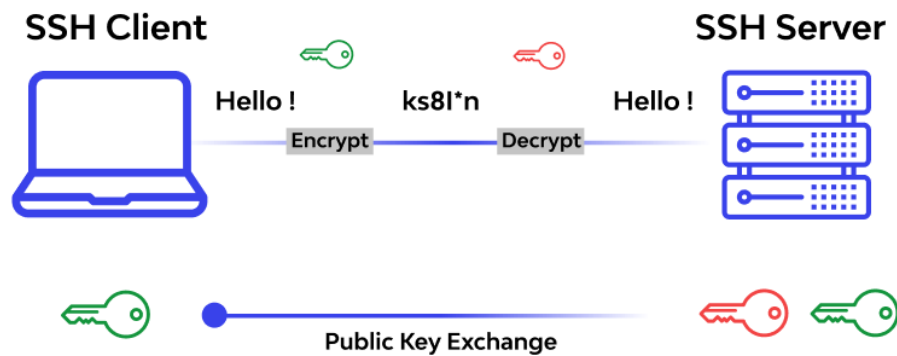
Gambar 9.8 Struktur Uniform Resource Locator (URL)

Telnet dan SSH

Telnet (*Terminal Network*) dan SSH (*Secure Shell*) merupakan dua buah protokol sekaligus aplikasi *client server* di internet yang umum digunakan untuk melakukan remote login ke komputer lain. SSH memiliki tingkat keamanan yang lebih tinggi dibanding telnet, sehingga sering digunakan untuk melakukan remote login ke komputer lain.

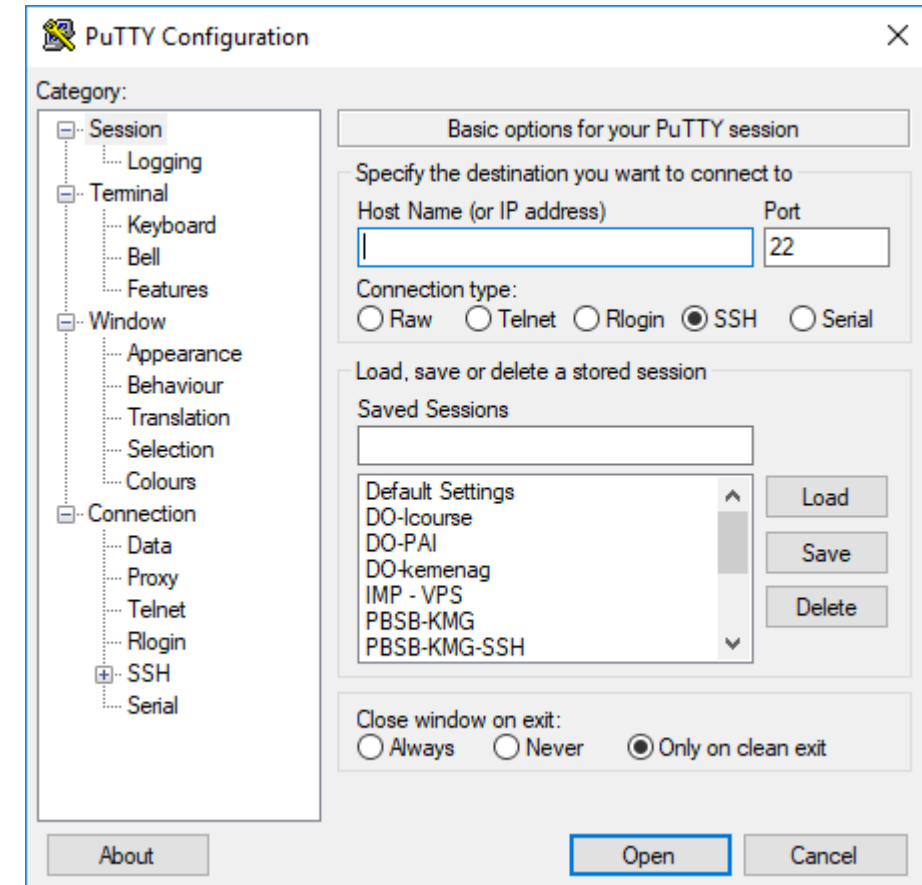


(a) Telnet (*Terminal Network*)



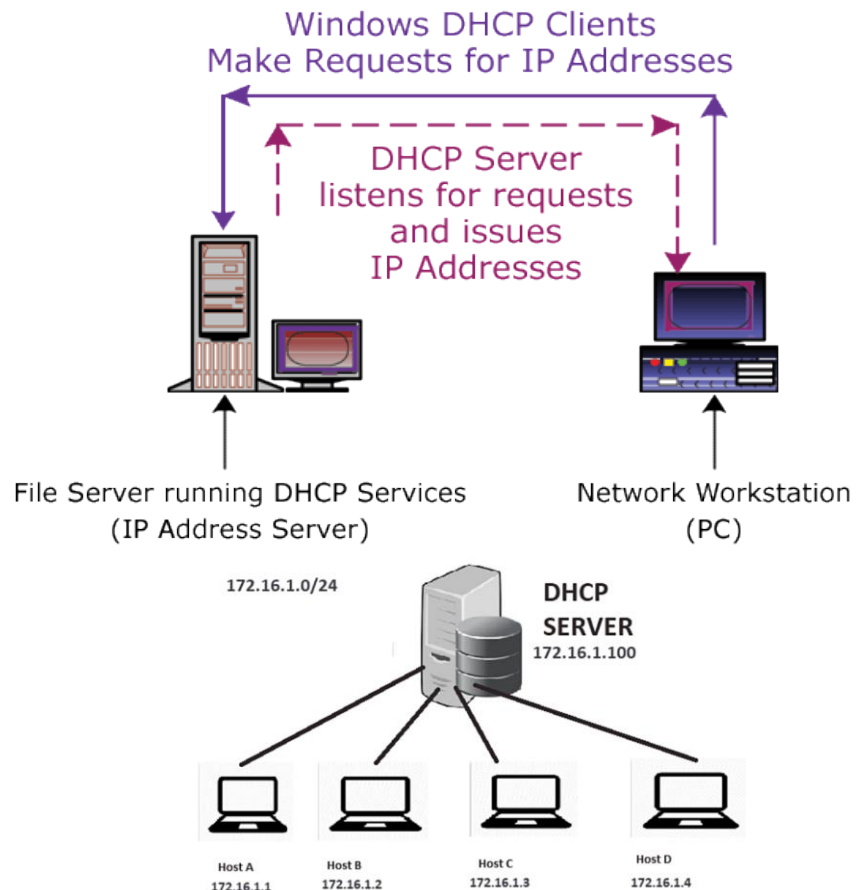
(b) SSH (*Secure Shell*)

Gambar 9.9 Penggunaan (a) Telnet (*Terminal Network*) dan (b) SSH (*Secure Shell*)



Gambar 9.10 Penggunaan Aplikasi Putty untuk melakukan remote login

DHCP (*Dynamic Host Configuration Protocol*)



Gambar 9.11 DHCP server

Merupakan salah protokol standar di jaringan komputer, yang berfungsi untuk membantu pengguna jaringan komputer memperoleh alamat (IP Address) secara cepat dan otomatis.

DHCP diciptakan agar dapat mudah terkoneksi ke jaringan komputer, tanpa harus repot melakukan konfigurasi alamat secara manual

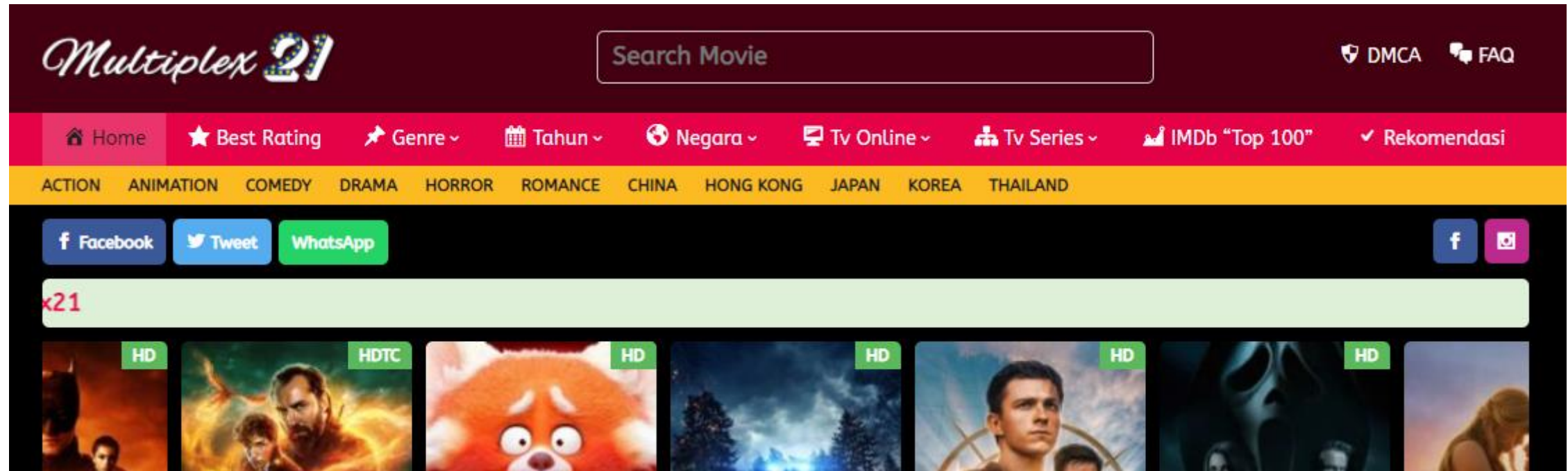


DNS (*Domain Named Server*)

DNS merupakan aplikasi standar di *aplication layer*, yang berfungsi untuk menterjemahkan pengalamatan, dari alamat IP ke alamat domain yang mudah untuk diingat oleh pengguna jaringan.

Sebagai contoh pengguna akan mudah mengingat alamat situs MULTIPLEX21 (dalam bentuk kalimat `www.MULTIPLEX21.com`), dibandingkan alamat IP MULTIPLEX21(dalam bentuk numerik `https://146.19.24.137/`).

www.multiplex21.com

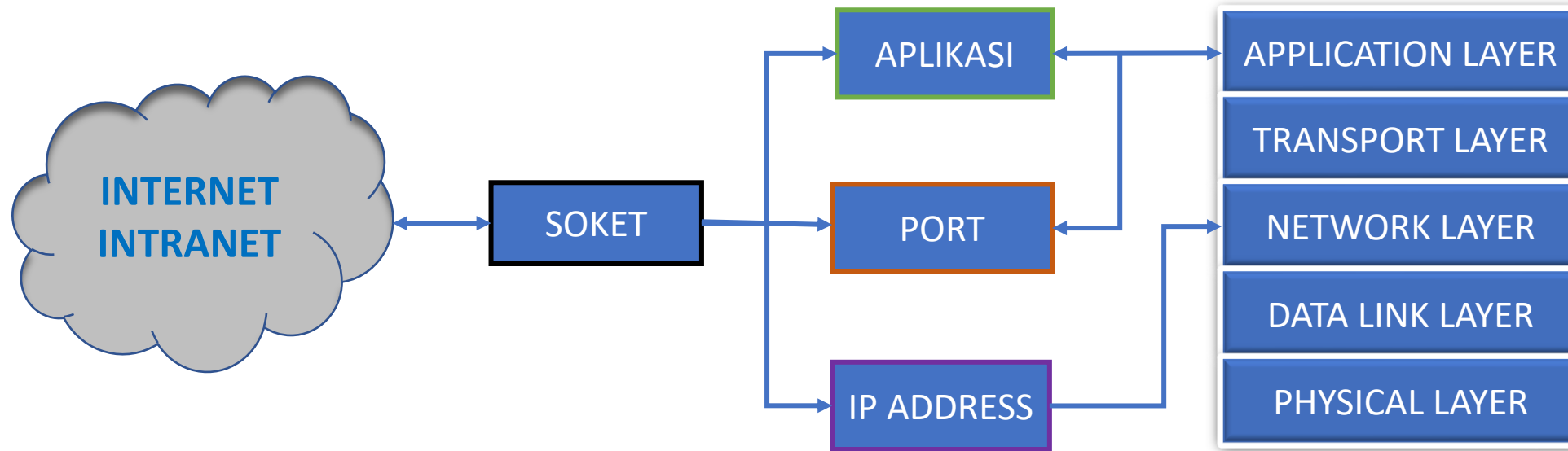


Gambar 9.12 Penggunaan DNS untuk membuka web Multiplex21



SOCKET PROGRAMMING

Soket didefinisikan sebagai sebuah *endpoint* (titik akhir) virtual yang digunakan untuk komunikasi antar komputer di jaringan. Soket inilah yang menghubungkan antara aplikasi web browser dengan jaringan komputer, untuk dapat mengakses situs tujuan dan menampilkannya dengan baik melalui aplikasi web browser.



Gambar 9.13 Ilustrasi soket serta kaitannya dengan aplikasi dan jaringan komputer

Terlihat bahwa untuk port aplikasi berjalan pada *Application Layer*, sedangkan pada alamat jaringan (IP Address) oleh *network layer* (IP Layer). *Socket* menjadi kunci untuk jalannya komunikasi antar komputer pada jaringan komputer melalui aplikasi yang digunakan oleh pengguna komputer. *Port* menjadi semacam nomor lokasi untuk jalur komunikasi melalui jaringan komputer. Sedangkan IP address merujuk kepada alamat dari komputer yang di tuju dan alamat komputer asal.



TERIMA
KASIH