*Program Studi D4 Teknik Informatika– Sekolah Vokasi Jl. Sari Asih No 54, Sarijadi, Kota Bandung, 40151*

# IDENTITAS

|  |  |  |
| --- | --- | --- |
| ***Bahan Kajian*** | | ***NPM :* 1214020** |
| * *Pengenalan Network Programming* * *Dasar-dasar Jaringan* * *Socket Programming Dasar* * *Protokol Transport - TCP* * *Protokol Transport – UDP* * *Multithreading* | | ***Nama :*  Farhan Rizki Maulana** |
| ***Kelas :* 4A** |
| ***Tandatangan :*** |
| ***Lama Pengerjaan*** | |  |
| 120 menit | |  |
| ***Jenis Pengerjaan (dipilih salah satu dengan cara digaris bawahi)*** | | |
| * ~~Tes Tulis~~ * ~~Tes Lisan (wawancara)~~ * ~~Praktek/Praktikum~~ | * ~~Presentasi (individu/kelompok)~~ * CBT (*Computer Based Test*) menggunakan LMS (*Learning Management System*) ULBI | |

**PETUNJUK PENGERJAAN**

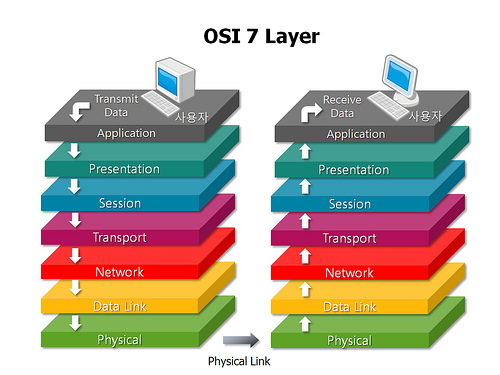
* Jawablah pertanyaan-pertanyaan soal berikut dengan baik dan benar
* Jawaban soal anda simpan sebagai file pdf dan submit di assignment e-learning sesuai batas waktu yang ditetapkan
* Proses koreksi akan melibatkan AI Detector, maka gunakan internet dengan bijak dan lakukan paraphrase dengan sebaik-baiknya. Nilai otomatis E jika AI Detector menghasilkan lebih dari 40%

*Program Studi D4 Teknik Informatika– Sekolah Vokasi Jl. Sari Asih No 54, Sarijadi, Kota Bandung, 40151*

# SOAL

1. Buatkan gambar alur kerja proses enkapsulasi dan dekapsulasi pada perangkat jaringan saat melakukan pengiriman data. Jelaskan prosesnya berdasarkan gambar ! **(BOBOT : 15)**

**Jawaban:**



Proses Enkapsulasi (Pengiriman Data).

1. Application Layer (Layer Aplikasi)

Pada layer aplikasi ini Data yang ingin dikirimkan oleh pengguna, seperti permintaan HTTP untuk membuka halaman Web yang disiapkan di aplikasi, seperti permintaan HTTP untuk mengakses halaman web dari browser. Data dari aplikasi ini diambil oleh aplikasi pengirim dan disiapkan untuk dikirim melalui jaringan.

1. Presentation Layer (Layer Presentasi)

Pada layer presentasi ini Data yang dihasilkan dari Application Layer perlu diproses lebih lanjut, seperti dikodean atau dienkripsi, Jika data memerlukan format khusus, seperti pengkodean UTF-8 atau enkripsi SSL/TLS, itu dilakukan di sini. Seperti mengonversi data HTTP menjadi format yang lebih aman misalkan melalui SSL/TLS. Data aplikasi dienkapsulasi lebih lanjut jika diperlukan missal enkripsi atau kompresi.

1. Session Layer (Layer Sesi)

Pada layer sesi ini ditugaskan untuk membuka, mengelola, dan menutup koneksi antara aplikasi yang berjalan di kedua perangkat yang terhubung, jika terdapat komunikasi berkelanjutan, sesi ini akan tetap aktif, seperti membuka sesi komunikasi untuk pertukaran data antara dua komputer misalnya melalui FTP dan HTTP. Layer ini menetapkan sesi komunikasi dan memberikan kontrol atas sesi yang sedang aktif.

1. Transport Layer (Layer Transport)

Pada layer transport ini Data dari Session Layer ini dibungkus menjadi segmen pada layer transport, menggunakan protokol seperti TCP atau UDP. Pada TCP, informasi seperti nomor urut, kontrol aliran, dan pengecekan kesalahan ditambahkan untuk memastikan data sampai dengan utuh dan urut. Seperti Protokol TCP menambahkan informasi seperti nomor urut dan checksum untuk memastikan data diterima tanpa kesalahan. Data dari sesi dikemas dalam segmen transport, yang berisi informasi kontrol untuk pengiriman yang andal.

1. Network Layer (Layer Jaringan)

Pada layer jaringan ini membungkus segmen transport ke dalam paket dan menambahkan alamatIPpengirimdanpenerima. Protokol yang digunakan adalah IP(Internet Protocol). Paket IP yang berisi alamat IP pengirim dan penerima. Segmen dari **Transport Layer** dibungkus dengan header IP, yang mencakup alamat IP sumber dan tujuan.

1. Data Link Layer (Layer Data Link)

Pada layer data link ini, paket IP dibungkus menjadi **frame**, dan alamat **MAC** pengirim dan penerima ditambahkan pada header frame. Protokol yang digunakan adalah Ethernet untuk komunikasi di jaringan lokal (LAN). Frame Ethernet dengan alamat MAC pengirim dan penerima. Paket IP dibungkus dalam frame Ethernet untuk ditransmisikan di jaringan lokal.

1. Physical Layer (Layer Fisik)

Pada layer fisik ini, Data akhirnya diubah menjadi **bit** yang bisa dikirimkan melalui media fisik seperti kabel tembaga, serat optik, atau melalui gelombang radio (untuk komunikasi nirkabel). Frame Ethernet diubah menjadi sinyal listrik, cahaya, atau gelombang radio. Frame data diubah menjadi bit yang akan ditransmisikan melalui media fisik (misalnya, kabel Ethernet).

Proses Dekapsulasi (Penerimaan Data).

1. Physical Layer (Layer Fisik)

Pada layer fisik ini, Data diterima dalam bentuk bit melalui media fisik seperti kabel atau gelombang radio dan diterjemahkan kembali menjadi frame. Frame data yang diterima diubah kembali menjadi format yang dapat dibaca di layer berikutnya.

1. Data Link Layer (Layer Data Link)

Pada layer data link ini, Frame data diperiksa untuk memastikan alamat **MAC** pengirim dan penerima sesuai. Jika valid, frame diteruskan ke layer jaringan. Header frame Ethernet dibuka, dan informasi MAC dibuang. Data diteruskan ke **Network Layer**.

1. Network Layer (Layer Jaringan)

Pada layer jaringan ini, Paket IP yang diterima diperiksa untuk memastikan alamat **IP** tujuan sesuai. Paket kemudian diteruskan ke **Transport Layer**. Header IP dibuka dan dibuang, hanya menyisakan segmen transport yang berisi data asli.

1. Transport Layer (Layer Transport)

Pada layer transport ini, Segmen yang diterima dari **Network Layer** diperiksa untuk kesalahan, nomor urut, dan kontrol aliran (pada TCP). Jika semua valid, data diteruskan ke **Session Layer**. Header TCP/UDP dibuang dan data diteruskan ke aplikasi.

1. Session Layer (Layer Sesi)

Pada layer sesi ini, Jika ada sesi aktif, sesi ini dikelola di layer ini untuk memastikan data diterima sesuai dengan urutan yang benar. Sesi yang sudah tidak diperlukan akan dihentikan.

1. Presentation Layer (Layer Presentasi)

Pada layer presentasi ini, Jika data dienkripsi atau dikompresi, proses dekripsi atau dekompresi dilakukan di sini. Data yang diterima diubah menjadi format yang bisa dimengerti oleh aplikasi.

1. Application Layer (Layer Aplikasi)

Pada layer aplikasi ini, Setelah melalui semua lapisan, data sampai di aplikasi penerima (seperti browser, email client, atau aplikasi lainnya). Data akhirnya diteruskan ke aplikasi yang membutuhkannya untuk diproses atau ditampilkan kepada pengguna.

1. Anda adalah seorang pekerja di sebuah perusahaan logistic sebagai pengelola jaringan. Rancanglah sebuah topologi jaringan sederhana untuk memastikan layanan perusahaan berjalan dengan optimal. Perusahaan ini memiliki perangkat-perangkat jaringan meliputi : 2 router (sebagai gateway utama), 2 switch, 1 hub, 1 Web Server (ujian.com), 1 DNS Server dan 8 unit komputer client. Perusahaan menggunakan IP address 192.168.80.0/28 untuk jaringan pada topologi ini. Melalui mekanisme subnetting, **A)** Tetapkan IP Address, Subnet Mask, IP Network, IP Broadcast dan IP Gateway untuk setiap perangkat. Khusus untuk komputer client, tambahkan IP DNS nya agar dapat mengakses ujian.com
   1. Untuk memudahkan pemetaan IP address gunakan table berikut :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama Perangkat** | **Interface** | **IP Address** | | | | | |
| **IP**  **Address** | **Subnet**  **mask** | **IP**  **Network** | **IP**  **Broadcast** | **IP**  **Gateway** | **IP DNS** |
| 1 | Router 1 | Interface 1 | 192.168.80.1 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 |  |
| 2 | Router 1 | Interface 2 | 192.168.80.2 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 |  |
| 3 | Router 2 | Interface 1 | 192.168.80.3 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 |  |
| 4 | Router 2 | Interface 2 | 192.168.80.4 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 |  |
| 5 | DNS Server | Interface 1 | 192.168.80.5 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 |  |
| 6 | Web Server | Interface 1 | 192.168.80.6 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 |  |
| 7 | Komputer Client 1 | Interface 1 | 192.168.80.7 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 8 | Komputer Client 2 | Interface 1 | 192.168.80.8 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 9 | Komputer Client 3 | Interface 1 | 192.168.80.9 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 10 | Komputer Client 4 | Interface 1 | 192.168.80.10 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 11 | Komputer Client 5 | Interface 1 | 192.168.80.11 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 12 | Komputer Client 6 | Interface 1 | 192.168.80.12 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 13 | Komputer Client 7 | Interface 1 | 192.168.80.13 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |
| 14 | Komputer Client 8 | Interface 1 | 192.168.80.14 | 255.255.255.240 | 192.168.80.0 | 192.168.80.15 | 192.168.80.1 | 192.168.80.5 |

* 1. Buatkan uraian analisis yang menjelaskan bagaimana komunikasi antar perangkat berjalan dengan lancar pada topologi diatas ! **(BOBOT : 70)**

Komunikasi antar perangkat pada topologi ini dapat berjalan dengan lancar dengan mengandalkan beberapa mekanisme dasar, seperti routing dan pengaturan alamat IP:

1. Router sebagai Gateway:

Router 1 dan Router 2 berfungsi sebagai gateway utama yang menghubungkan perangkat dalam jaringan lokal (LAN) dengan jaringan lain. Mereka akan mengatur pengiriman paket data antar perangkat melalui IP address dan memfasilitasi routing data dari satu perangkat ke perangkat lain.

1. Pengalamatan IP dan Subnetting:

Alamat IP yang dialokasikan memastikan bahwa perangkat dalam jaringan ini dapat saling berkomunikasi, baik itu antara komputer client, DNS server, maupun web server. Subnet mask yang digunakan memastikan bahwa setiap perangkat dalam jaringan ini terhubung dalam satu subnet yang sama, memungkinkan komunikasi yang efisien.

1. DNS dan Web Server:

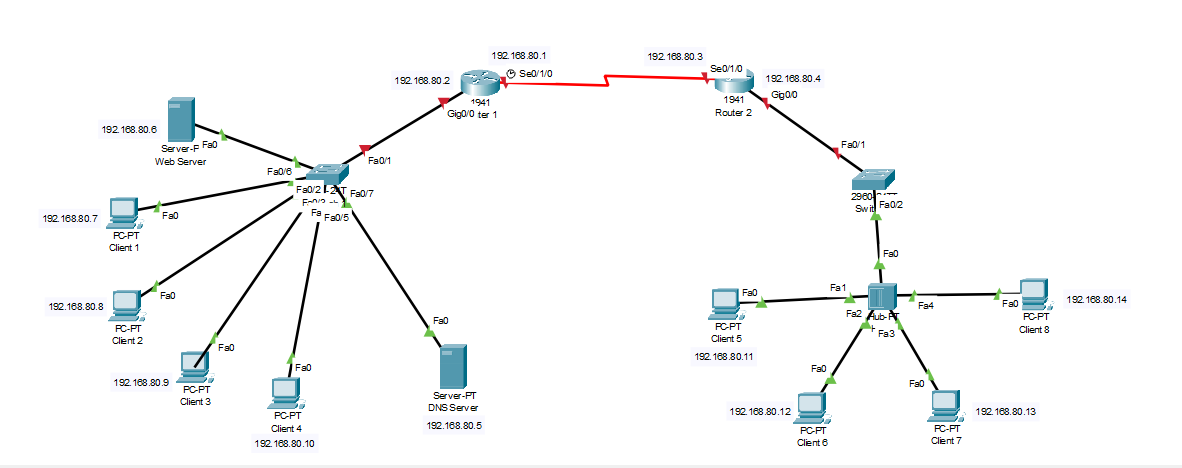
Komputer client akan mengirimkan permintaan DNS ke DNS server (192.168.80.5) untuk mencari alamat IP web server (ujian.com). DNS server akan mengonversi nama domain menjadi alamat IP yang dapat digunakan untuk mengakses web server.

1. Routing dan Pemetaan Paket:

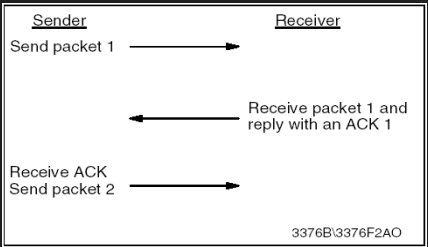
**Router 1** dan **Router 2** memastikan paket data dikirimkan dengan benar menggunakan alamat **IP Gateway** yang telah ditetapkan. Proses routing antara router akan memastikan bahwa data sampai ke tujuannya, baik itu untuk perangkat lain di jaringan lokal atau ke luar jaringan melalui koneksi internet.

*Program Studi D4 Teknik Informatika– Sekolah Vokasi Jl. Sari Asih No 54, Sarijadi, Kota Bandung, 40151*

1. Gambarkan cara kerja TCP dan UDP serta buatkan penjelasannya kaitkan dengan topologi yang anda buat pada soal no.2 **(BOBOT : 10)**

**Konsep Cara Kerja TCP dan UDP**

1. TCP (Transmission Control Protocol)



TCP adalah protokol **berorientasi koneksi** yang menyediakan komunikasi yang andal antar dua perangkat. TCP memastikan bahwa data yang dikirim antara pengirim dan penerima sampai dengan **utuh** dan **urut**, serta tidak ada data yang hilang. TCP juga mengatur **kontrol aliran**, **penanganan kesalahan**, dan **pengaturan ulang paket** yang hilang.

Proses Kerja TCP:

1. Establishing Connection (Proses 3-Way Handshake):

* Sebelum data dapat dikirimkan, **TCP** melakukan proses **3-way handshake** untuk membentuk koneksi antara pengirim dan penerima.
* **PC 1 (192.168.80.7)** akan mengirimkan **SYN** ke **Web Server (192.168.80.6)** untuk memulai koneksi.
* **Web Server** akan merespons dengan **SYN-ACK**.
* **PC 1** akan mengirimkan **ACK** untuk menyelesaikan koneksi dan memastikan kedua perangkat siap mengirim dan menerima data.

1. Data Transmission:

* Setelah koneksi terbentuk, data akan dikirim dalam bentuk **segmen TCP** yang terurut. Setiap segmen dilengkapi dengan **nomor urut** dan **checksum** untuk memastikan data sampai dengan benar.
* **PC 1** mengirimkan segmen data kepada **Web Server**, dan setiap segmen yang diterima oleh server akan dikirimkan kembali dengan **acknowledgment** untuk memastikan bahwa segmen tersebut diterima dengan benar.

1. Reliability:

* Jika **Web Server** tidak menerima segmen data dengan benar (misalnya karena segmen hilang atau rusak), server akan meminta pengiriman ulang segmen tersebut ke **PC 1**. **TCP** menjamin bahwa data sampai dengan **utuh** dan **terurut**.

1. Connection Termination:

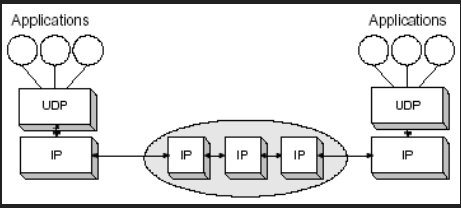
* Setelah data berhasil ditransfer, koneksi TCP ditutup dengan proses **4-way handshake**, di mana kedua perangkat mengirimkan pesan **FIN** untuk mengakhiri sesi komunikasi.

Contoh dalam Topologi:

* Misalnya, **PC 1 (192.168.80.7)** ingin mengakses halaman web pada **Web Server (192.168.80.6)**.
* **PC 1** akan mengirimkan **SYN** untuk memulai koneksi TCP dengan **Web Server**.
* **Web Server** akan merespons dengan **SYN-ACK**, dan **PC 1** mengirimkan **ACK** untuk membentuk koneksi.
* Setelah koneksi terbentuk, **PC 1** akan mengirimkan permintaan data, dan **Web Server** akan mengirimkan respons yang dibalut dalam **segmen TCP**.
* Seluruh data akan diperiksa dengan **checksum** dan **nomor urut** untuk memastikan data sampai dengan benar.

1. UDP

UDP adalah protokol **tanpa koneksi** yang lebih cepat daripada TCP, tetapi tidak menjamin pengiriman data yang **andalan**. UDP tidak melakukan pemeriksaan kesalahan, kontrol aliran, atau pengaturan ulang paket yang hilang. Oleh karena itu, UDP sering digunakan untuk aplikasi yang memerlukan **kecepatan tinggi** dan dapat mentolerir kehilangan data, seperti **streaming video/audio**, **gaming online**, atau **DNS query**.



Proses Kerja UDP:

1. Connectionless Communication:

* **PC 1 (192.168.80.7)** mengirimkan data dalam bentuk **datagram UDP** ke **Web Server (192.168.80.6)** tanpa melakukan **handshake**.
* UDP langsung mengirimkan data tanpa memastikan apakah penerima siap atau tidak.

1. No Acknowledgment:

* Setelah data dikirim, **PC 1** tidak menunggu **acknowledgment** dari **Web Server**. Ini artinya, tidak ada proses untuk memastikan bahwa data telah diterima dengan benar.
* Jika datagram hilang atau rusak dalam perjalanan, tidak ada mekanisme untuk meminta pengiriman ulang.

1. Faster Transmission:

* Karena tidak ada proses handshaking atau pengiriman ulang data, UDP lebih **cepat** daripada TCP, yang sangat ideal untuk aplikasi real-time.

Contoh dalam Topologi :

* Misalnya, **PC 1 (192.168.80.7)** mengirimkan permintaan DNS ke **DNS Server (192.168.80.5)**.
* **PC 1** mengirimkan permintaan DNS menggunakan **UDP**. Proses ini sangat cepat, karena tidak ada mekanisme pengiriman ulang atau konfirmasi penerimaan data.
* **DNS Server** mengirimkan kembali IP address untuk **Web Server** menggunakan UDP. Jika ada datagram yang hilang dalam perjalanan, UDP tidak akan meminta pengiriman ulang.

1. Dalam konteks pemrograman jaringan berbasis python, ide apa yang mungkin anda bisa lakukan untuk mendukung implementasi aktivitas apapun pada topologi jaringan yang anda buat diatas! **(BOBOT : 5)**

**Aplikasi Web Server dengan Python dan Flask**

Tujuan dan Implementasi :

* **Web Server** ini dibuat dengan menggunakan **Python** dan **Flask** untuk memberikan **layanan web** yang dapat diakses oleh perangkat lain di jaringan yang sama (seperti **PC 1**, **PC 2**, dll.) melalui **HTTP**.
* Web Server ini berfungsi untuk menangani **permintaan (requests)** dari pengguna, memberikan **respon** (misalnya halaman web atau data), dan memungkinkan interaksi antara **PC Client** dan **Web Server** dalam topologi jaringan yang telah Anda buat sebelumnya.

Kaitan dengan Topologi:

* Dalam topologi yang Anda buat, **Web Server (192.168.80.6)** akan melayani **permintaan HTTP** dari **PC Client (192.168.80.7)** yang ingin mengakses halaman web.
* **Web Server** ini akan digunakan oleh perangkat-perangkat di jaringan untuk mengakses **halaman web**, seperti halaman utama atau halaman **ujian.com**.