Implementasi VPN WireGuard untuk Secure Remote Access

Kelompok 7

1. Janumelah (2201020048)
2. Mohd Allifyan Baitul Nesam (2201020023)
3. Anjas Revaldo (2201020108)
4. Safitri Wulandari (2201020085)
5. Danish Arya Yudhistira (2201020110)
6. M. Aditya Egi Dwi Nata (2201020141)

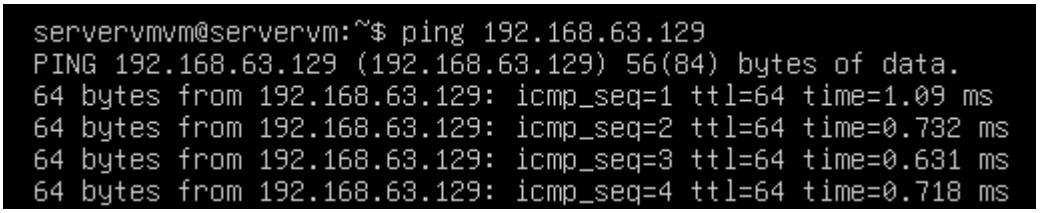
Minggu 5: Capture Wireshark sebelum & sesudah VPN

Pada minggu 5 dilakukan proses pengujian keamanan komunikasi jaringan dengan melakukan packet capture menggunakan Wireshark. Tujuan tahap ini adalah membandingkan kondisi lalu lintas jaringan sebelum menggunakan VPN (non-encrypted traffic) dan sesudah VPN aktif (encrypted traffic), sehingga dapat dibuktikan bahwa WireGuard berhasil memberikan enkripsi pada komunikasi antar mesin.

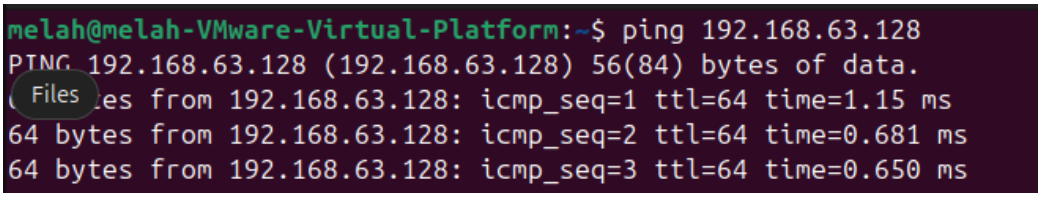
Tahapan dan Proses:

1. Melakukan ping antar mesin sebelum VPN

   Pada tahap ini dilakukan pengujian konektivitas dasar antara kedua mesin (server dan client) dengan menggunakan perintah ping melalui jaringan lokal sebelum VPN diaktifkan. Tujuannya adalah memastikan bahwa kedua VM dapat saling terhubung secara langsung tanpa enkripsi, serta menjadi dasar pembanding sebelum dilakukan pengujian melalui WireGuard VPN.

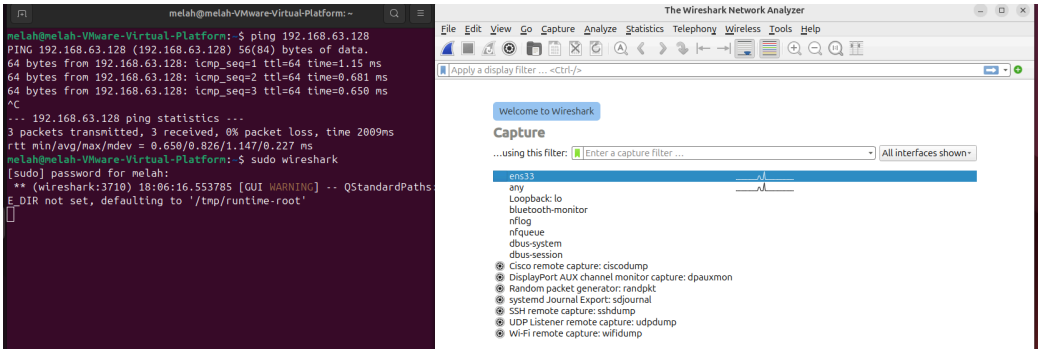


   Pada sisi server, dilakukan pengujian konektivitas dengan menjalankan perintah ping menuju alamat IP client (192.168.63.129). Hasil output menunjukkan respons ICMP yang stabil dengan waktu latensi rendah, menandakan bahwa server dapat terhubung ke client melalui jaringan lokal tanpa hambatan. Pengujian ini memastikan bahwa jalur komunikasi dasar antara kedua mesin sudah berjalan sebelum VPN diaktifkan.



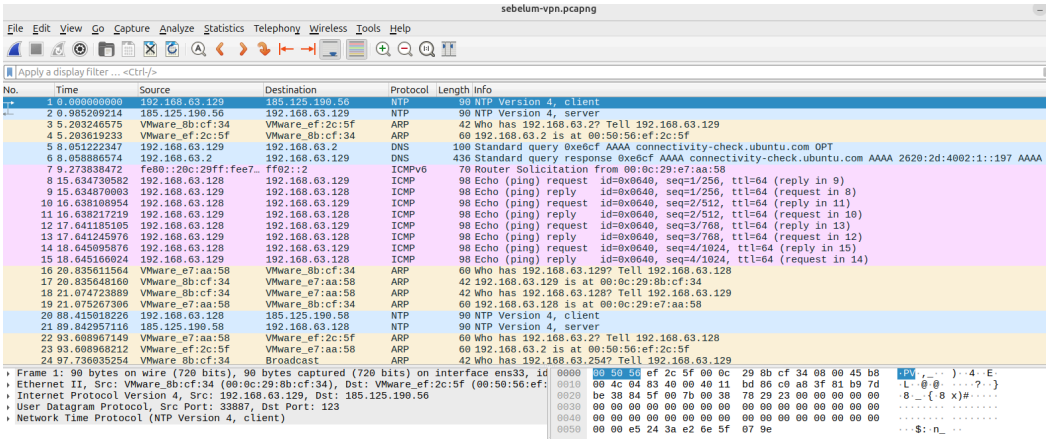
   Pada sisi client, dilakukan pengujian serupa dengan mengirimkan paket ICMP ke alamat IP server (192.168.63.128). Hasil ping menunjukkan balasan yang konsisten dan tidak ada packet loss, sehingga membuktikan bahwa client dapat menjangkau server secara langsung.

2. Membuka aplikasi Wireshark untuk melakukan *packet capture*



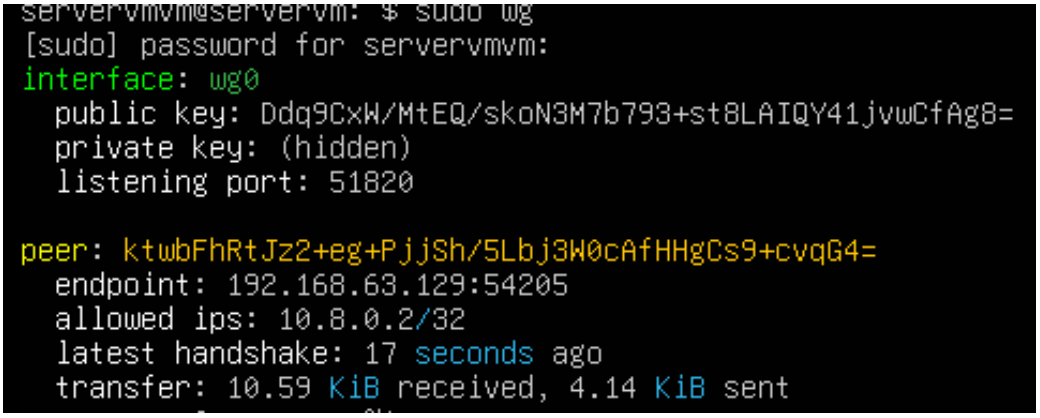
Pada tahap ini, client membuka aplikasi Wireshark menggunakan perintah sudo wireshark untuk memulai proses packet capture. Setelah aplikasi berjalan, Wireshark menampilkan daftar interface jaringan yang tersedia seperti ens33 yang akan digunakan untuk merekam lalu lintas jaringan sebelum dan sesudah VPN diaktifkan.

3. *Capture* lalu lintas jaringan sebelum VPN



Pada tahap ini dilakukan proses packet capture menggunakan Wireshark sebelum VPN WireGuard dijalankan. Hasil capture memperlihatkan bahwa seluruh paket jaringan seperti ICMP (ping), DNS query, ARP, dan NTP ditampilkan dalam bentuk plaintext, sehingga seluruh informasi sumber, tujuan, dan payload dapat terbaca dengan jelas. Kondisi ini menunjukkan bahwa komunikasi antar mesin masih berjalan tanpa enkripsi, sehingga rentan untuk disadap jika melalui jaringan publik.

4. Melakukan verifikasi *handshake* VPN WireGuard



Pada sisi server, perintah sudo wg digunakan untuk menampilkan status koneksi WireGuard. Output menunjukkan bahwa server telah mengenali peer (client) melalui publik key yang sesuai, serta menampilkan informasi endpoint client dan allowed IP. Bagian "latest handshake" menunjukkan waktu terakhir kedua perangkat berhasil bertukar kunci dan membentuk sesi terenkripsi. Data transfer (received/sent) juga mulai bertambah, menandakan bahwa komunikasi VPN sudah aktif.

Pada sisi client, perintah yang sama (sudo wg) memperlihatkan bahwa client berhasil terhubung dengan server. Terlihat informasi publik key server, endpoint server, allowed IP, serta nilai "latest handshake" yang mengonfirmasi bahwa pertukaran kunci terjadi dengan sukses. Nilai transfer data juga mulai muncul, menandakan bahwa client telah berhasil mengirim dan menerima paket melalui tunnel VPN.

5. Melakukan verifikasi *routing* di Client

```
melah@melah-VMware-Virtual-Platform:~$ ip route
default via 192.168.63.2 dev ens33 proto dhcp src 192.168.63.129 metric 100
10.8.0.0/24 dev wg0 proto kernel scope link src 10.8.0.2
192.168.63.0/24 dev ens33 proto kernel scope link src 192.168.63.129 metric 100
```

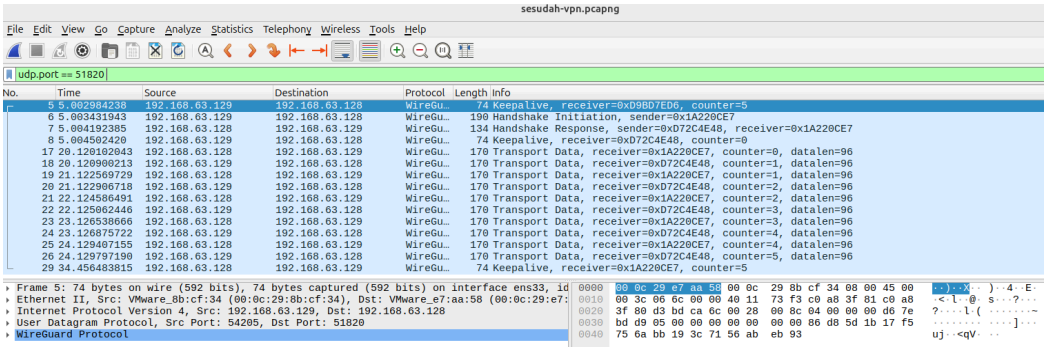
Pada tahap ini client menjalankan perintah ip route untuk memastikan bahwa rute jaringan untuk interface VPN (wg0) sudah terbentuk. Hasilnya menunjukkan adanya rute menuju jaringan 10.8.0.0/24 melalui interface wg0, dengan alamat sumber 10.8.0.2. Ini menandakan bahwa client sudah terhubung ke tunnel VPN dan dapat mengirim trafik ke server VPN melalui jalur tersebut.

6. Melakukan ping IP VPN dari Server

```
servervmvm@servervm:~$ ping 10.8.0.2
PING 10.8.0.2 (10.8.0.2) 56(84) bytes of data.
64 bytes from 10.8.0.2: icmp_seq=1 ttl=64 time=2.35 ms
64 bytes from 10.8.0.2: icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from 10.8.0.2: icmp_seq=3 ttl=64 time=1.38 ms
64 bytes from 10.8.0.2: icmp_seq=4 ttl=64 time=1.22 ms
64 bytes from 10.8.0.2: icmp_seq=5 ttl=64 time=1.23 ms
```

Server melakukan pengujian konektivitas ke client melalui perintah ping 10.8.0.2. Hasil ping menunjukkan balasan (reply) dengan waktu respon sangat rendah, menandakan bahwa tunnel VPN telah bekerja dengan baik dan komunikasi antara server dan client melalui jaringan privat berjalan tanpa masalah.

7. *Capture* lalu lintas jaringan setelah VPN



Gambar di atas menunjukkan tampilan Wireshark saat melakukan packet capture setelah koneksi VPN WireGuard diaktifkan. Pada daftar paket terlihat bahwa seluruh komunikasi jaringan ditandai dengan protokol WireGuard (Wg), yang terdiri

dari beberapa tipe paket seperti Handshake Initiation, Handshake Response, Keepalive, serta Transport Data.

Paket-paket Transport Data menampilkan informasi seperti receiver, counter, dan datalen, namun tidak menampilkan konten asli data karena seluruh payload telah dienkripsi. Hal ini menjadi bukti bahwa setelah VPN aktif, lalu lintas jaringan tidak lagi terbaca sebagai data mentah (non-encrypted traffic), melainkan hanya terlihat sebagai paket terenkripsi oleh WireGuard.

Hasil:

1. Berhasil mendapatkan packet capture trafik sebelum VPN sebagai data pembanding.
2. Capture setelah VPN menunjukkan lalu lintas yang sepenuhnya terenkripsi dengan WireGuard.
3. Pembuktian berhasil bahwa komunikasi melalui WireGuard tidak dapat dibaca, sesuai tujuan project.
4. Data capture siap digunakan untuk dokumentasi pada minggu ke-6.