**Worksheet 3**

Nama : Samuel Ludwig Ian  
NPM : 1806191471  
Kelas : Jarkomdat B

Link yang diakses :

* Apache.org/index.html
* Facebook.com/

**A. Explain the flow of HTTP persistent in your experiment**

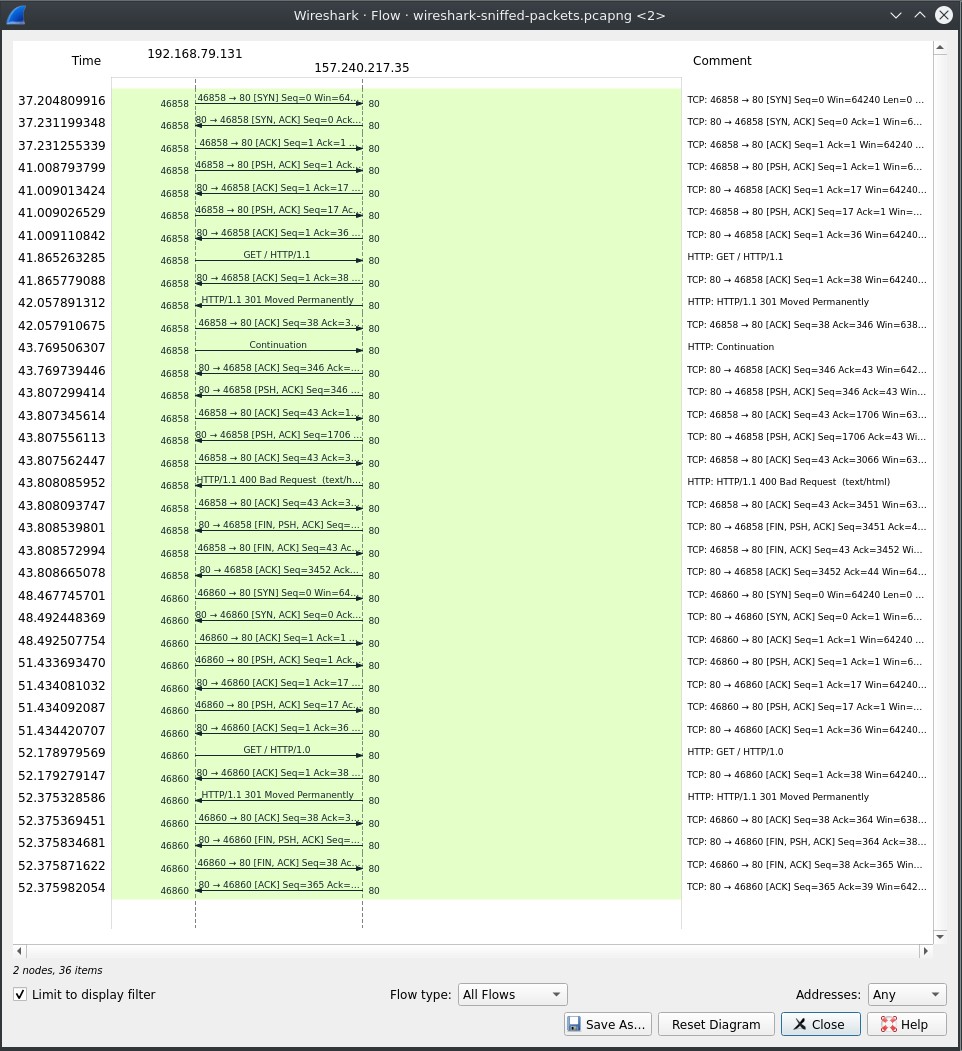
Dalam eksperimen, saya menemukan adanya penanda dimana Connection yang diberikan adalah keep alive. Dalam melakukan koneksi kepada server, dilakukan **syn**chronization yang berasal dari kita (remote) kepada tujuan (server) yang ditandai dengan flag [SYN]. Setelah itu, server menjawab dengan adanya **ack**nowledgement atas **syn**chronization request yang ditandai dengan flag [SYN, ACK]. Selanjutnya, computer kita (remote) memberikan **ack**nowledgement juga dengan mengirimkan flag [ACK]. Setelah dilakukan **syn**chronization, didapati bahwa kita mengirimkan (mem**p**u**sh**) data dan telah berhasil menerima data (**ack**nowledged) dengan adanya flag [PSH, ACK]. Komunikasi ini dilakukan selama beberapa kali dengan arah bolak-balik.

Saat kita melakukan request GET melalui protocol HTTP/1.1 dengan menjalankan perintah “GET / HTTP/1.1”, server menjawab request kita dengan konfirmasi (**ack**nowledgement) yang ditandai oleh flag [ACK]. Kemudian, server merespon dengan protocol yang sama yaitu HTTP/1.1 dan memberi tahu bahwa resource yang mau kita akses sudah dipindahkan sehingga sebaiknya dilakukan redirect dengan adanya informasi “301 Moved Permanently”. Komputer kita mendapatkan pesan ini dan memberikan flag [ACK] untuk konfirmasi.

Karena komputer kita maupun server belum memberikan/mengirim data yang ditandai dengan flag [PSH], maka dikeluarkan flag **Continuation** yang menandakan bahwa dilakukan pengiriman data yang besar sehingga harus dipecah menjadi beberapa paket. Setelah itu, komputer dan server masih terus melakukan pengiriman (**p**u**sh**) dan konfirmasi penerimaan data (**ack**nowledgement) agar koneksi tetap tersambung dan tidak di alokasikan untuk resource lain yang ditandai dengan adanya flag [PSH, ACK] dan [ACK].

Saat dilakukan pemutusan koneksi dengan menekan tombol ctrl+c, dikeluarkan infomasi **fin**ish yang di kirimkan (**p**u**sh**)dengan flag [FIN, PSH, ACK] **ack**nowledgement disini digunakan sebagai penanda konfirmasi dari pengiriman data sebelumnya. Setelah komputer menerima informasi bahwa server ingin memutus koneksi TCP yang tadi dibuat, maka komputer mengkonfimasi (**ack**nowledgement) dan mengirimkan informasi bahwa komputer juga ingin memutus koneksi (**fin**ish)dengan mengirimkan flag [FIN, ACK]. Koneksi TCP ditutup dengan adanya konfirmasi (**ack**nowledgement) dari server yang menerima flag [FIN] yang dikirimkan komputer.

**B. Explain the differences of HTTP persistent and HTTP non persistent**

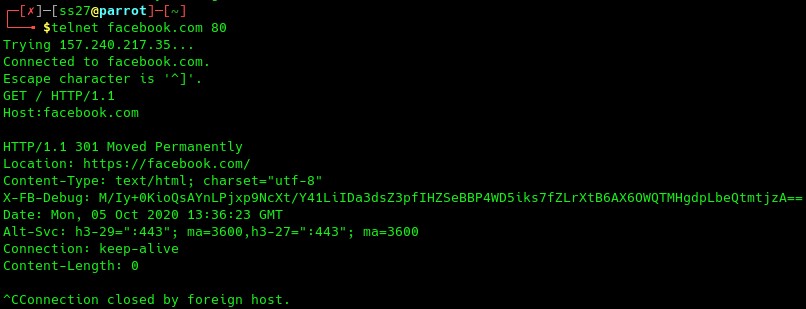
*Note: Saya menaruh garis merah sebagai penanda, bagian atas adalah HTTP Persistent dan bagian bawah adalah HTTP non-persistent*

HTTP persistent dan HTTP non-persistent dapat dibedakan berdasarkan jenis koneksinya. HTTP persistent mempertahankan dan menunggu koneksi TCP setelah dilakukan request, namun HTTP non-persistent memutuskan koneksi TCP setelah request selesai direspon. HTTP non-persistent dapat diidentifikasi saat melakukan koneksi TCP melalui telnet yang ditandai oleh header , sedangkan HTTP persistent dapat diidentifikasi dengan cara yang sama dengan jenis koneksi berbeda yakni . Selain itu, untuk melakukan koneksi persistent, saya melakukan koneksi TCP ke facebook.com dengan HTTP/1.1 dan untuk koneksi non-persistent, saya melakukan koneksi TCP ke facebook.com juga, namun dengan HTTP/1.0. Karena koneksi persistent mengalokasikan resource secara khusus untuk suatu komunikasi, maka koneksi dapat menjadi penuh dan resource menjadi macet, oleh karena itu umumnya digunakan HTTP/1.1 untuk persistent connection karena kecepatan HTTP/1.1 yang lebih tinggi dibanding dengan HTTP/1.0.

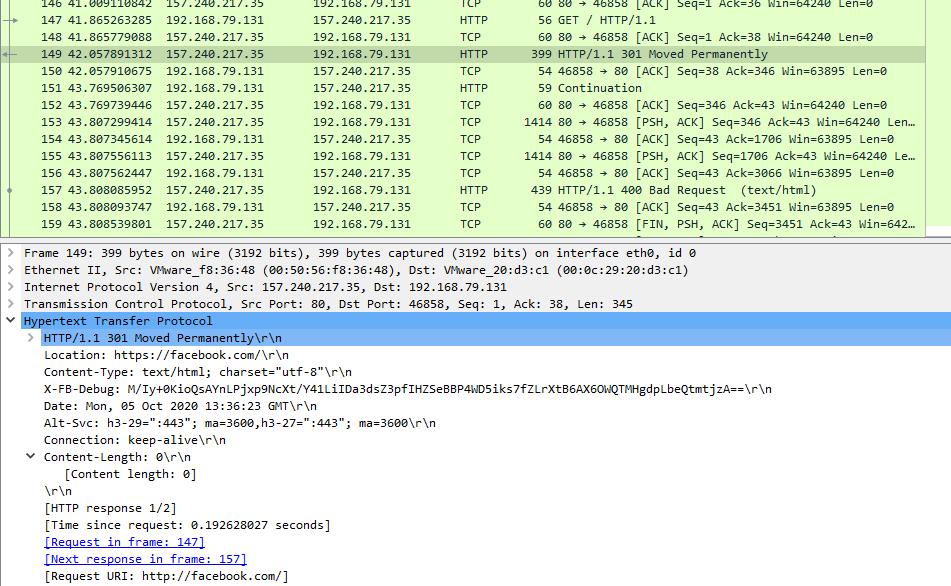
Untuk flow dari penggunaan non-persistent communication, proses berjalan seperti berikut: Client melakukan koneksi TCP → Request dari client→ Response diberikan server → Koneksi TCP diputuskan → Client melakukan koneksi TCP → Request dari client→ Response diberikan server → Koneksi TCP diputuskan → dst. Hal ini disebabkan oleh adanya pemutusan koneksi setelah request telah direspon oleh server pada HTTP non-persistent. Berbeda dengan HTTP non-persistent, HTTP persistent tetap membiarkan komunikasi TCP tersambung setelah request dari client telah di respon, sehingga flow atau prosesnya berjalan seperti berikut: Client melakukan koneksi TCP → Request dari client→ Response diberikan server → Request dari client→ Response diberikan server → … → Koneksi TCP diputuskan.

**C. Bukti screenshot pengerjaan tugas**

1. Link : facebook.com/  
   Port : 80  
   Request line : GET / HTTP/1.1  
   Status line : HTTP/1.1 301 Moved Permanently  
   Header lines : (Lihat gambar)  
   Body/data : Server tidak memberikan

   
*Note: Untuk nomor 1*

**HEADER Lines**

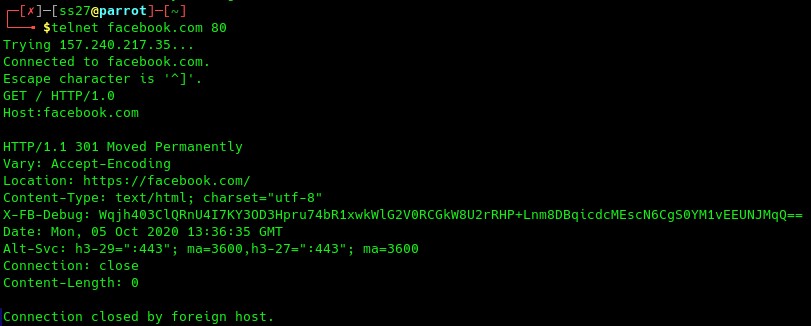
   
*Note: Untuk nomor 1*

**HEADER Lines**

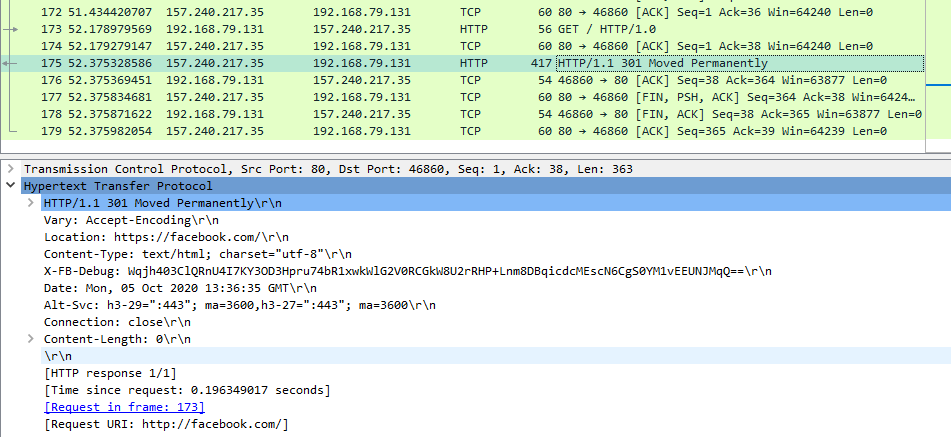
**Status Lines**

**Request Lines**

1. Link : facebook.com/  
   Port : 80  
   Request line : GET / HTTP/1.0  
   Status line : HTTP/1.1 301 Moved Permanently  
   Header lines : (Lihat gambar)  
   Body/data : Server tidak memberikan

   
*Note: Untuk nomor 2*

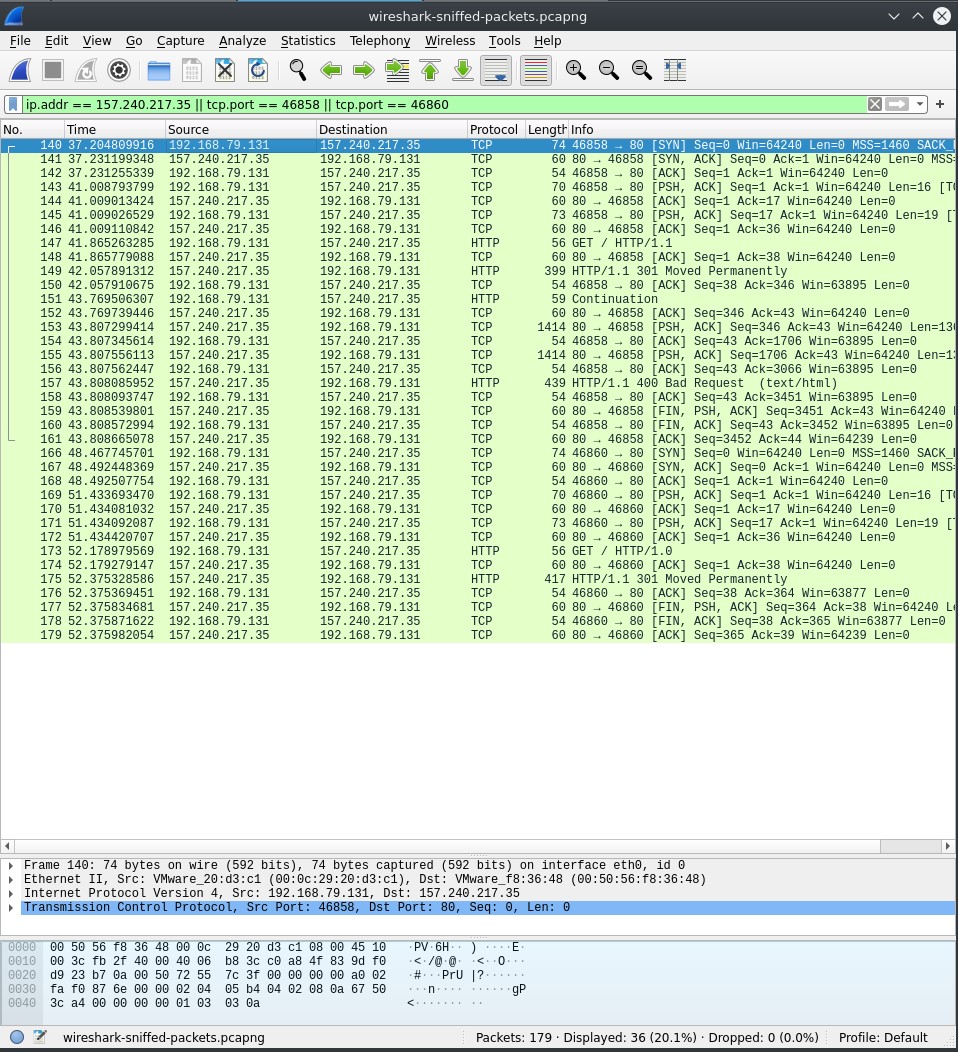
**HEADER Lines**

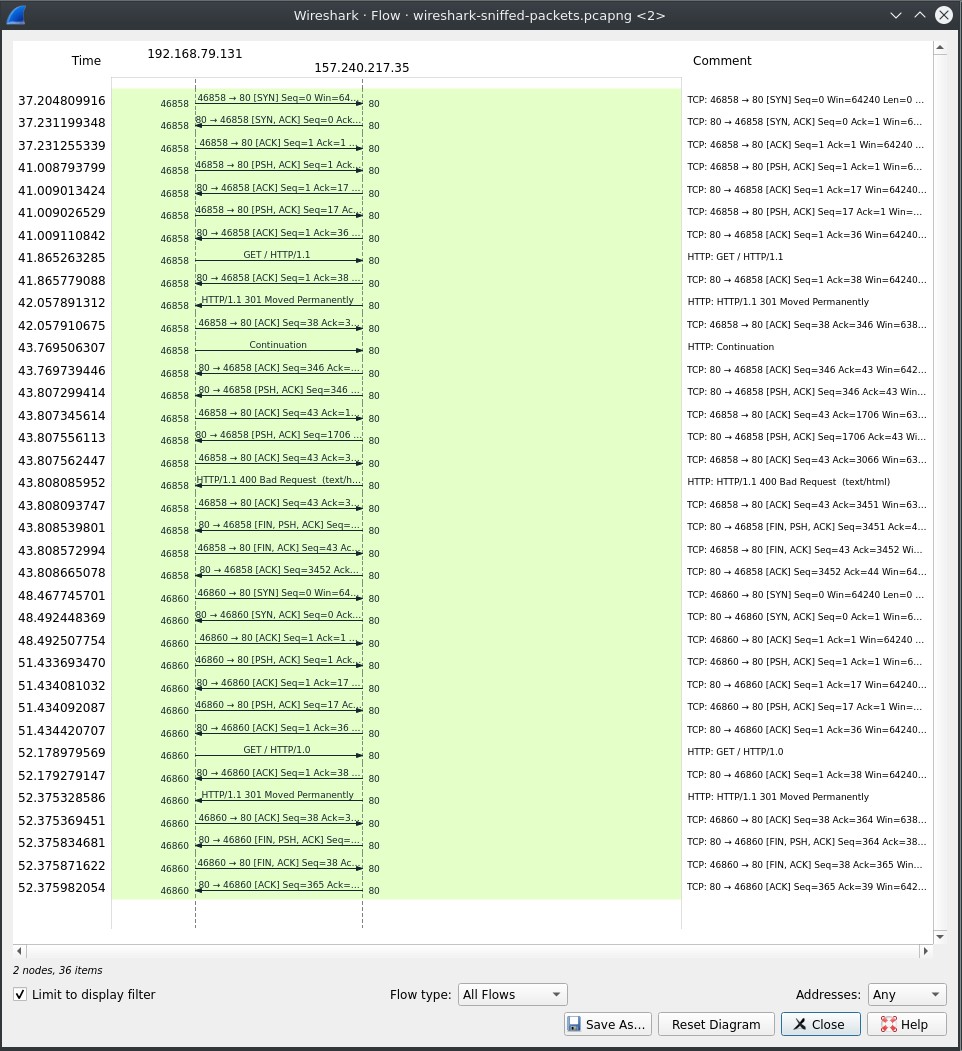
  
*Note: Untuk nomor 2*

**HEADER Lines**

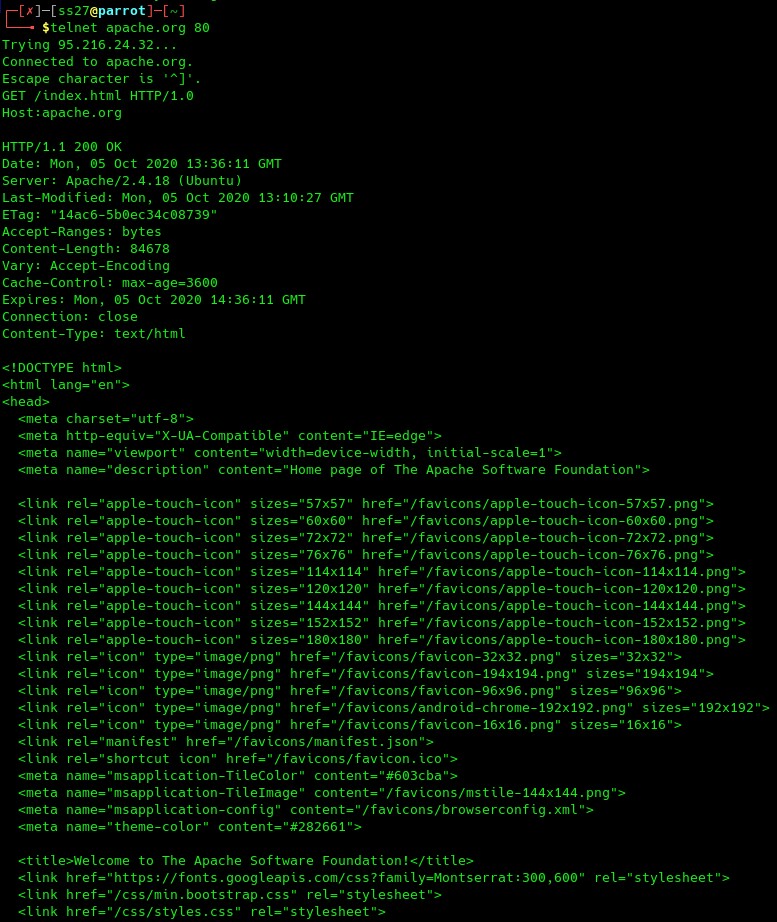
**Status Lines**

**Request Lines**

  
*Note: Hasil filter untuk nomor 1 dan 2.  
Saya menaruh garis merah sebagai penanda, bagian atas adalah HTTP Persistent dengan request menggunakan HTTP/1.1 dan bagian bawah adalah HTTP non-persistent dengan menggunakan HTTP/1.0*

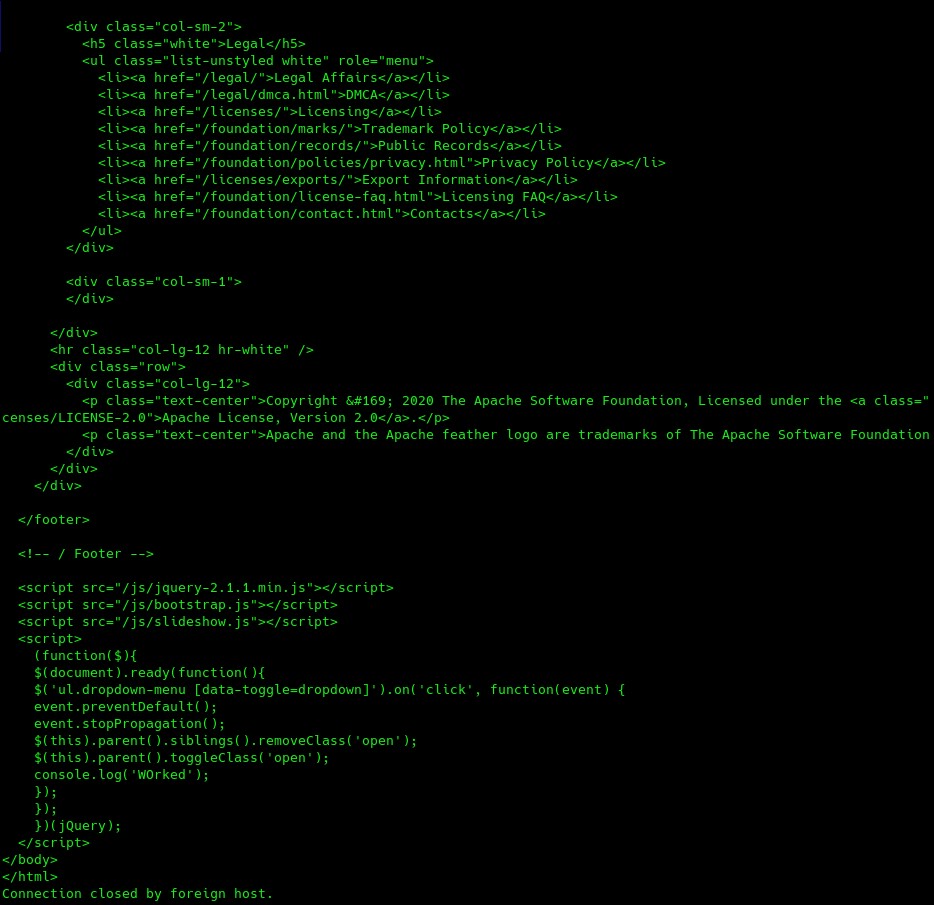
  
*Note: Flow Graph untuk nomor 1 dan 2  
Saya menaruh garis merah sebagai penanda, bagian atas adalah HTTP Persistent dengan request menggunakan HTTP/1.1 dan bagian bawah adalah HTTP non-persistent dengan menggunakan HTTP/1.0*

1. Link : apache.org/index.html  
   Port : 80  
   Request line : GET /index.html HTTP/1.0  
   Status line : HTTP/1.1 200 OK  
   Header lines : (Lihat gambar)  
   Body/data : (Lihat gambar)

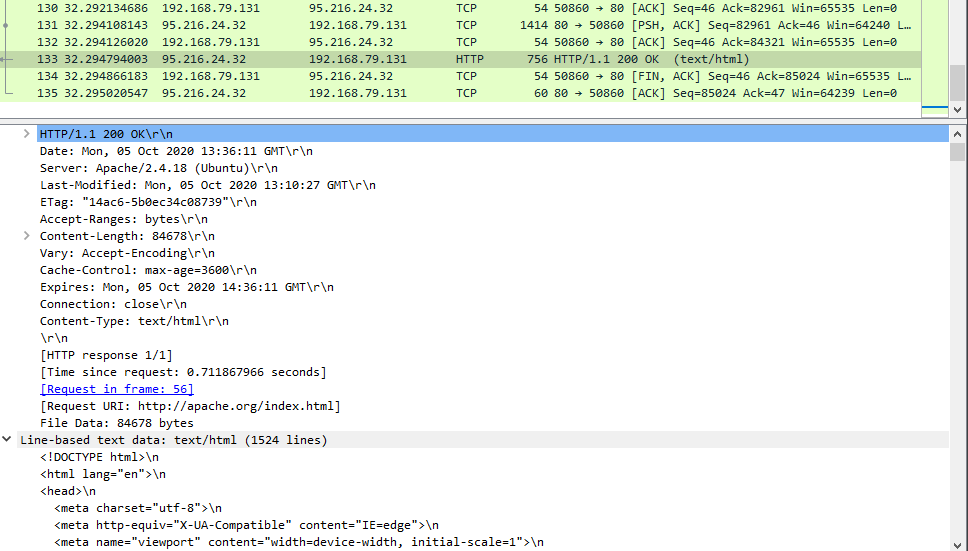
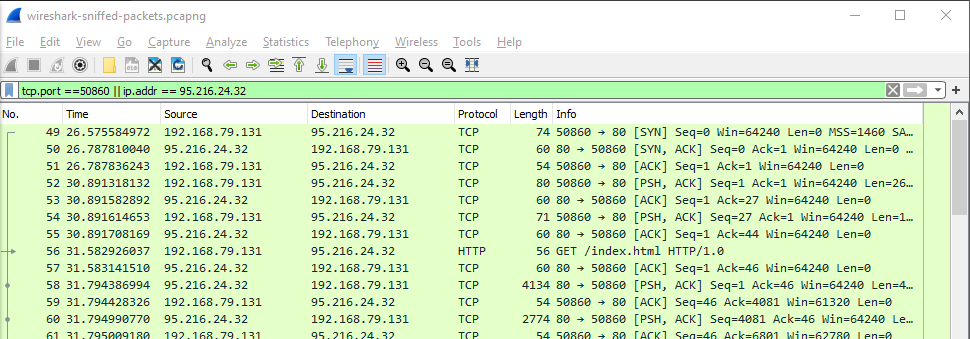
   
*Note: Untuk nomor 3*

**HEADER Lines**

**Body/data**

   
*Note: Untuk nomor 3*

**Body/data**

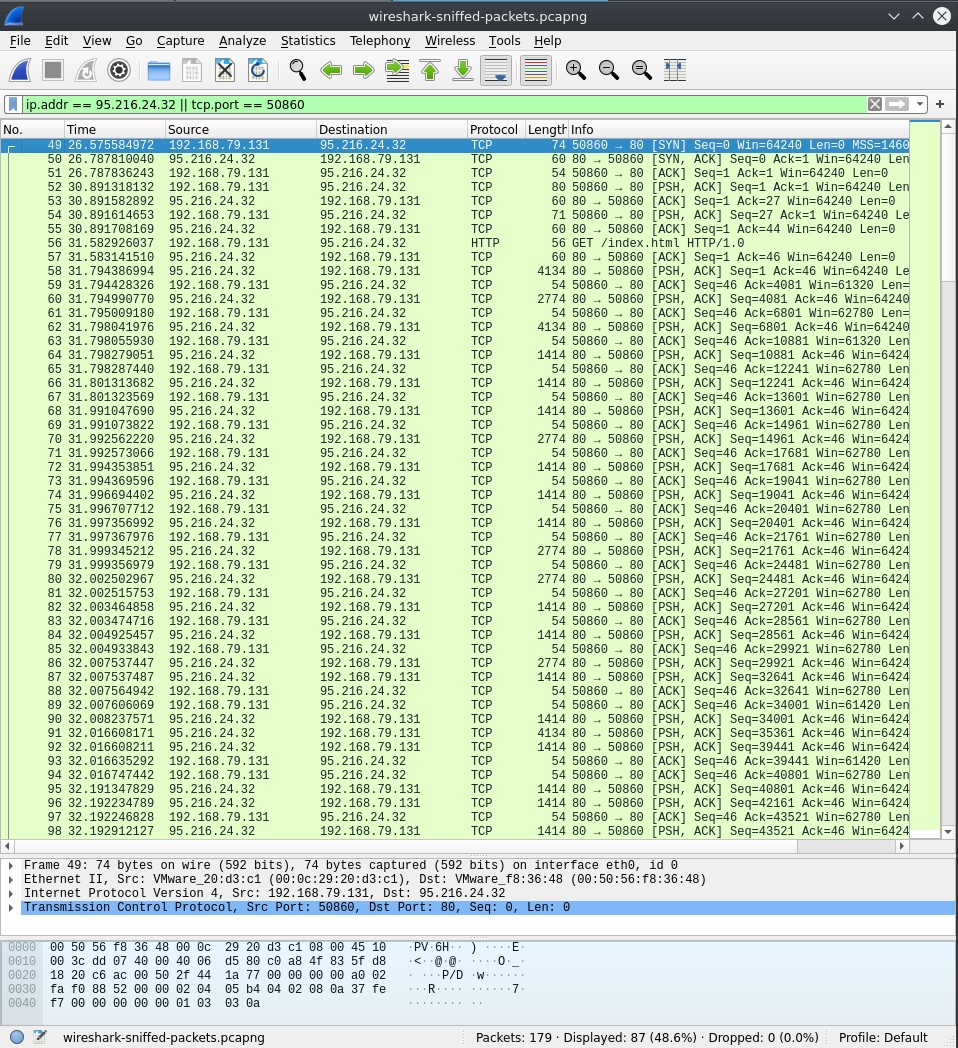
   
*Note: Untuk nomor 3, html tidak di screenshot semuanya karena terlalu panjang*

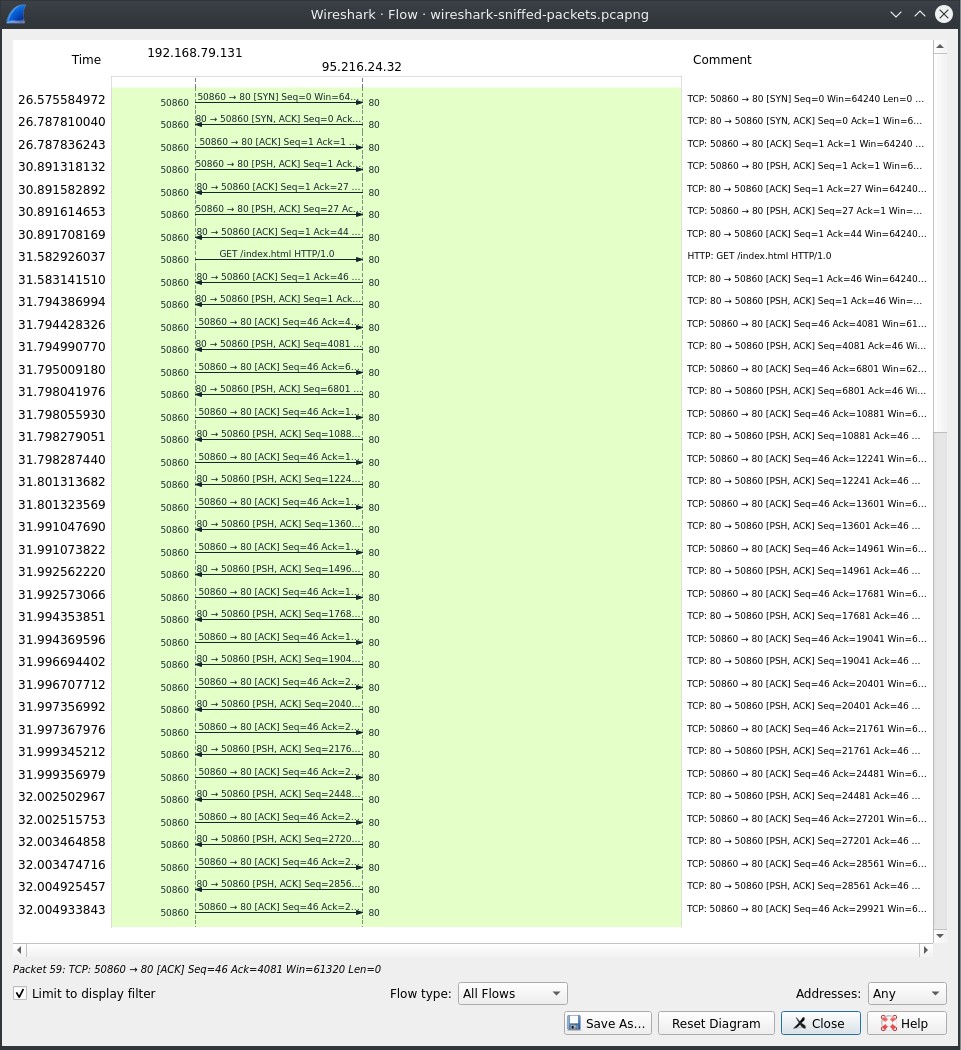
**Body/data**

**HEADER Lines**

**Status Lines**

**Request Lines**

*Note: Hasil filter untuk apache.org.*

   
*Note: Hasil Flow Graph untuk apache.org.*

**D. Sources**

<https://linuxhint.com/http_wireshark/>

<https://osqa-ask.wireshark.org/questions/13384/display-http-header#:~:text=Wireshark%20captures%20full%20packets%20by,was%20processed%20by%20the%20proxy>.

<https://www.the-art-of-web.com/system/telnet-http11/>

<https://unix.stackexchange.com/questions/237635/using-telnet-to-get-website-header/237648>

<https://www.juniper.net/documentation/en_US/junos/topics/topic-map/tcp-configure-features.html#:~:text=The%20FIN%20flag%20indicates%20the,recipient%2C%20depending%20on%20the%20OS>.

<http://www.freekb.net/Article?id=938>