BAB II
LANDASAN TEORI

2.1 Jaringan Komputer

Jaringan komputer disusun sebagai *layer* yang dibangun di atas satu sama lain. Tujuan dari tumpukan *layer* ini adalah digunakan untuk mengurangi rancangan jaringan dan mengimplementasikan perangkat lunak jaringan yang sulit. Setiap *layer* mengimplementasikan beberapa layanan tertentu dan menyediakannya untuk tingkat *layer* yang lebih tinggi. Setiap *layer* memiliki beberapa protokol, hanya satu protokol per *layer* yang digunakan untuk koneksi *end-to-end* (Tanenbaum, 2003, Bab 1.3).

2.1.1 OSI Reference Model

Model referensi OSI (Open System Interconnection) adalah model konsep bertujuan sebagai konsep komunikasi dari sistem telekomunikasi tanpa memperhatikan struktur dan teknologi internal, referensi OSI dibuat dan dikembangkan oleh International Standards Organization (ISO). Model referensi OSI bertujuan untuk membuat perbedaan antara tiga konsep utama arsitektur jaringan: layanan, antarmuka, dan protokol. Layanan menentukan tugas di setiap layer, antarmuka menentukan bagaimana cara mengakses layanan, dan protokol adalah jaringan yang melakukan jenis layanan yang sebenarnya. Model ini hanya sebagai jenis fungsi dimana setiap layer harus penuh, bukan protokol yang tepat untuk digunakan. OSI terdiri dari tujuh *layer*, seperti yang ditunjukkan pada Tabel 2.1 (Tanenbaum, 2003, Bab 1.4).

Tabel 2.1 Layer dari model referensi OSI

7	Application Layer
6	Presentation Layer
5	Session Layer
4	Transport Layer
3	Network Layer
2	Data Link Layer
1	Physical Layer

2.1.2 TCP/IP Reference Model

Transmision Control Protocol/Internet Protocol (TCP/IP) adalah protokol koneksi yang menyediakan layanan transfer data byte-stream yang memberikan hasil yang sama pada percobaan berulang. Sangat sedikit asumsi untuk kelebihan layanan transfer data yang mendasarinya dibawah *layer* TCP, pada dasarnya bahwa TCP dapat memperoleh layanan datagram sederhana dan berpotensi tidak dapat diandalkan melalui protokol tingkat bawah.

Model referensi TCP/IP pada awalnya dirancang sejak ARPANET (Advanced Research Projects Agency Network). Nama model berasal dari dua protokol utamanya, yaitu TCP dan IP. Nama tersebut juga menyiratkan fakta bahwa protokol diciptakan sebelum model, sehingga membuat tidak cocok untuk tumpukan protokol lainnya. Dibandingkan dengan model OSI, model TCP/IP gagal untuk membedakan dengan tiga konsep yang jelas disebutkan dalam bagian model TCP/IP digambarkan dalam Tabel 2.2 (Tanenbaum, 2003, Bab 1.4).

Tabel 2.2 Layer dari model referensi TCP/IP

4	Application Layer
3	Transport Layer
2	Internet Layer
1	Host-to-Network Layer

2.1.3 Physical layer

Semua jaringan dan *layer* selanjutnya bedasarkan pada physical layer. Tugas physical layer adalah mentransmisikan data biner ke bit, melewati saluran komunikasi (Tanenbaum, 2003, Bab 2). Physical layer akan menangani sebagai berikut:

a. Konversi data digital menjadi sinyal listrik yang menyesuaikan, dan sebaliknya,
b. Pembentukan dan penghancuran koneksi,
c. Masalah waktu dan spesifikasi fisik lainnya

2.1.4 Data link layer

Layer berikutnya adalah *Layer* data link yang berkaitan dengan mendapatkan *frame* dari satu ujung *link* komunikasi ke ujung lainnya. *Layer* data link memiliki banyak tugas, salah satu tugas penting adalah sebagai berikut (Tanenbaum, 2003, Bab 3):

a. **Framing**: *Layer* data link membagi aliran bit yang diterimanya dari *layer* fisik menjadi urutan bingkai. Tujuan dari proses ini adalah untuk memungkinkan deteksi

dan koreksi kesalahan transmisi yang mungkin, misalnya dengan memberikan jumlah yang dihitung untuk setiap frame yang dikirim atau dengan menambahkan beberapa bit yang berlebihan.

b. **Kesalahan kendali**: Dalam hal ini layanan yang dapat diandalkan, berorientasi koneksi seperti TCP, semua frame yang dikirim oleh *host* pertama harus dikirim dengan berhasil dalam urutan yang tepat ke *layer* jaringan host kedua. Sambutan berhasil diterima oleh frame dan pengiriman ulang frame yang rusak atau hilang akan dilakukan di *layer* data link.

c. **Jalur kendali**: Mekanisme untuk mencegah pengirim dari membebani penerima dengan *frame* yang masuk. Mekanisme dapat didasarkan pada umpan balik dari penerima atau dibangun di protokol yang digunakan. Kontrol aliran dilakukan juga di *layer* atas.

2.1.5 Network Layer

Network layer bertanggung jawab untuk mengirimkan datagram sepanjang jalur dari sumber ke tujuan. Proses transfer paket melalui titik (node) jaringan menengah dan mungkin melalui beberapa jaringan terpisah sebelum mencapai tujuan yang disebut dengan *routing*. Banyak jaringan yang sama dengan berbagai ragam protokol yang berbeda. Menghubungkan beberapa jaringan dengan router disebut internetworking. *Layer* ini menyediakan layanan tanpa sambungan, usaha terbaik menuju *layer* transportasi.

Internet menggunakan berbagai macam protokol yang terkait dengan *layer* jaringan, tetapi jelas yang paling penting dari itu adalah Internet Protocol (IP). Ini adalah standar dari World Wide Web (WWW) dan banyak aplikasi jaringan lainnya, serupa dengan e-mail, instant messenger, multimedia, dan aplikasi *peer-to-peer*. Selain itu, *layer* jaringan juga mencakup jalur protokol dan beberapa protokol kendali, seperti ICMP (Dynamic Host Configuration Protocol), ARP (Address Resolution Protocol), dan DHCP (Dynamic Host Conguration Protocol) (Tanenbaum, 2003, Bab 5).

2.1.6 Transport Layer

Fungsi utama transport layer adalah untuk membangun koneksi *end-to-end* antara proses aplikasi, di mana data ditukarkan dalam bentuk segmen. Transport layer sangat mirip dengan *layer* jaringan dan kebanyakan layanan mereka pada dasarnya sama (misalnya alamat, penanganan aliran dan kesalahan). Perbedaan utamanya adalah perangkat lunak transport layer

berjalan pada perangkat yang dipakai pengguna, dan perangkat lunak *layer* jaringan sebagian besar berjalan pada router dan perangkat jaringan lainnya. Tujuannya adalah untuk meningkatkan layanan yang disediakan layer network sebaik mungkin, memastikan kualitas layanan, dan memberikan kesempatan untuk memiliki koneksi yang paling andal dan rawan kesalahan.

Internet memiliki dua protokol transport utama: *UDP* dan *TCP*. *UDP* cocok untuk beberapa interaksi klien-server dan aplikasi multimedia secara langsung, tetapi *TCP* digunakan oleh aplikasi yang menuntut keandalan, seperti transfer file dan email. Komunikasi *TCP* dan *UDP* dilakukan melalui titik akhir yang disebut soket. Host menggunakan alamat *IP* dan nomor port untuk mengarahkan segmen ke soket yang sesuai. Nomor port di bawah 1024 adalah port yang umum dicadangkan untuk layanan standar (Tanenbaum, 2003, Bab 6).

2.1.7 Application Layer

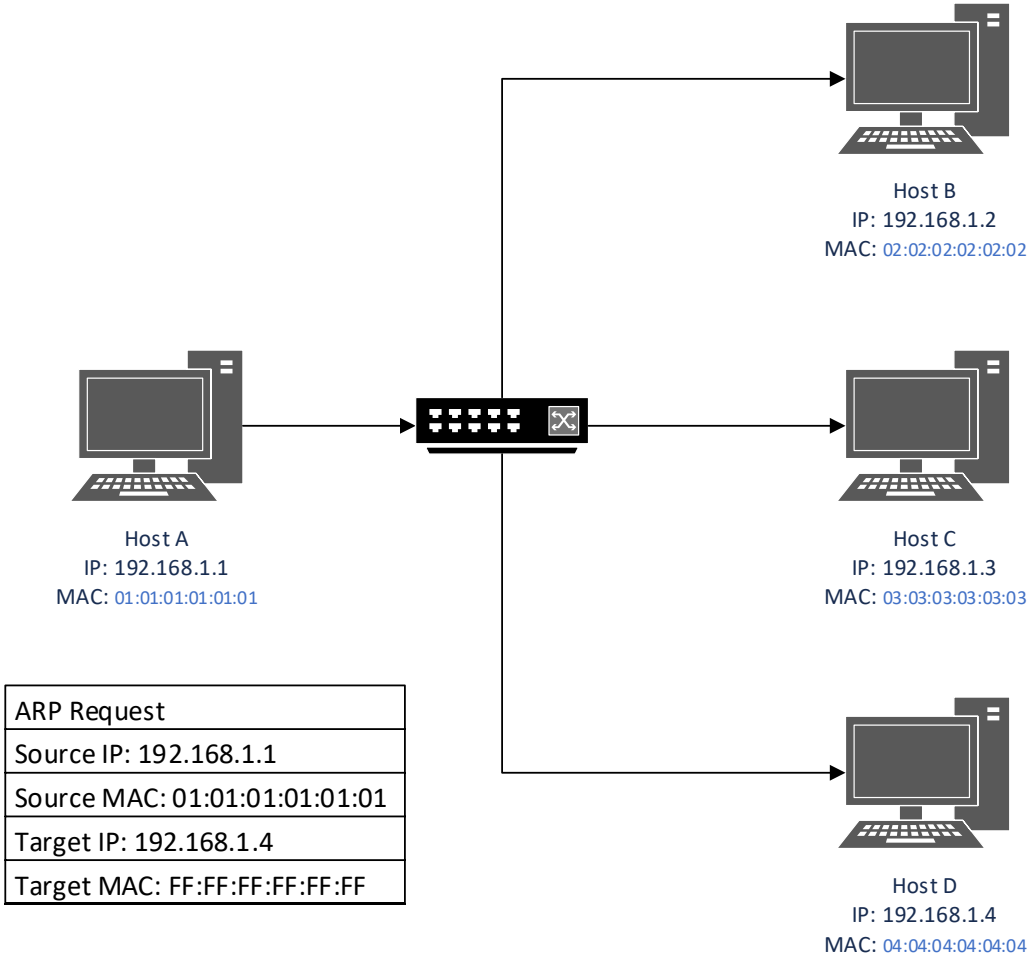
Application Layer melakukan interaksi langsung dengan aplikasi jaringan yang menerapkan fungsi yang dijalankan oleh pengguna. Aplikasi pada host menghubungkan komunikasi dengan aplikasi lain pada host dengan cara mengirim pesan satu sama lain bersama *layer* dasar yang menangani pesan transfer yang sebenarnya. Protokol Application layer mendefinisikan pesan sintaksis dan pesan semantik yang dipertukarkan di antara aplikasi.

Di internet menawarkan banyak aplikasi yang bermanfaat, tetapi mampu disebut dua yang paling penting adalah e-mail dan World Wide Web (WWW). WWW adalah sebuah sistem terstruktur dari hypertext yang saling berhubungan, yang dapat berisi banyak jenis konten seperti file, gambar, audio dan video. Sumber daya yang sering diakses melalui Uniform Resource Locator (URL), yang bagian utamanya adalah nama domain. Nama domain adalah karakter identifikasi yang mendefinisikan dalam ranah oronomi administrasi yang digunakan oleh manusia untuk memudahkan akses internet, bukan alamat IP yang sulit diingat. DNS yang menangani penyelesaian nama domain ke alamat IP (Tanenbaum, 2003, Bab 7).

2.1.8 Komunikasi End-To-End Antar Host

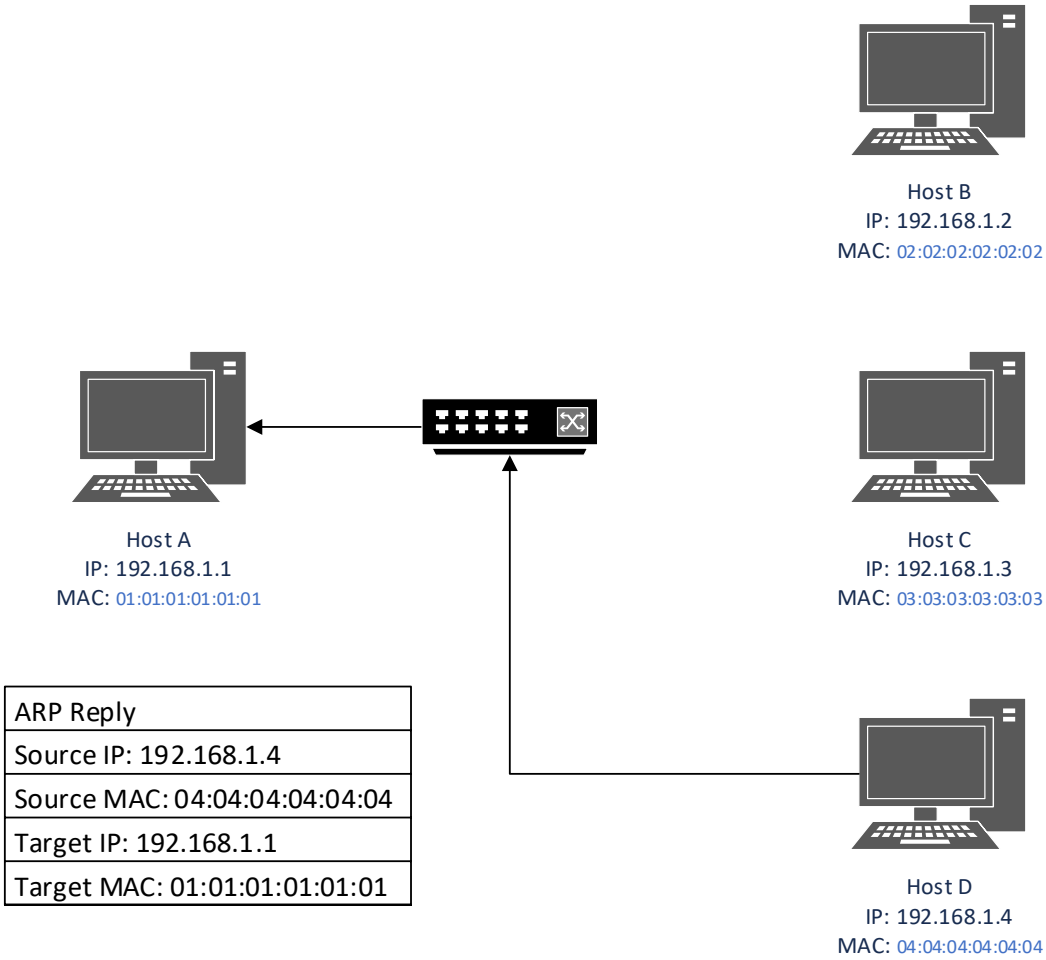
Dalam jaringan komputer, host A mengirim paket ke salah satu host, pertama – tama memeriksa cache ARP (Address Resolution Protocol) untuk pemetaan alamat yang sesuai. Jika host A tidak tahu alamat MAC dari host D, maka ia akan mengirim permintaan ARP ke setiap host (*broadcast*) di jaringan LAN. Permintaan ARP berisi alamat IP destinasi yang diketahui

secara publik dan meminta alamat MAC host destinasi yang sesuai. Proses permintaan ARP diilustrasikan dalam Gambar 2.1 (Tanenbaum, 2003).



Gambar 2.1 Contoh ARP request

Host memiliki alamat IP tertentu, dalam hal ini host D akan merespons host A dengan jawaban ARP yang berisi alamat MAC-nya, seperti yang ditunjukkan pada gambar 2.2. selanjutnya host A dan D berkomunikasi dengan menggunakan alamat MAC masing – masing, setiap paket yang dikirim masih dilakukan *broadcast* ke seluruh LAN, tetapi Network Interface Card (NIC) dari suatu host memeriksa tujuan alamat MAC dari paket dari paket tersebut sebelum menerimanya. Namun, paket sniffer memiliki kemampuan untuk mengatur NIC dalam mode *promiscuous*, di mana cara tersebut bekerja sebagai menerima seluruh paket yang mengalir melalui segmen jaringan ditunjukkan pada gambar 2.2 (Qadeer dkk., 2010; S. Ansari dkk., 2002; Sanders, 2011).



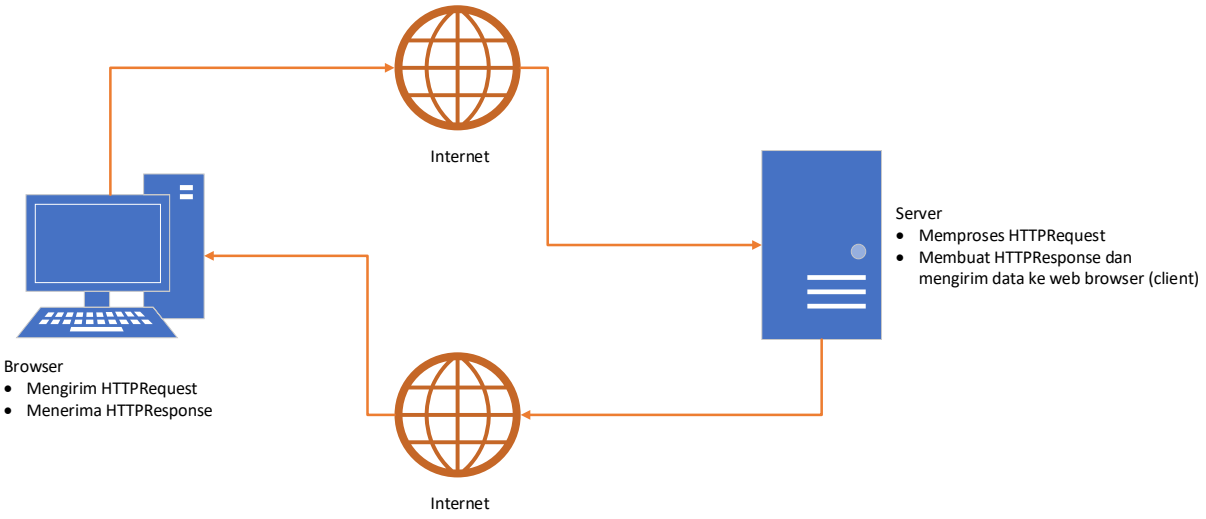
Gambar 2.2 Contoh ARP reply

2.2 Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) adalah protokol application-level untuk distribusi, kolaborasi, dan media system informasi. HTTP telah digunakan sejak World Wide Web diciptakan pada tahun 1990. Versi pertama pada HTTP adalah HTTP/0.9, adalah versi sederhana untuk transfer data mentah melintas ke Internet. Komunikasi HTTP antara *server* dan *client*, yaitu oleh permintaan (Request) dan respons sebagai berikut:

a. Client mengirimkan permintaan HTTP ke web.
b. Web server akan menerima permintaan.
c. Server akan menjalankan aplikasi untuk memproses permintaan.
d. Kemudian server akan mengirim respons ke web browser untuk client.

Dari empat langkah tersebut, akan mengulang terus menerus tergantung pengguna yang akan mengunjungi situs web yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Sirkulasi protokol HTTP dalam internet.

Pada Gambar 2.4, web *server* memiliki fitur *patch*, dimana fitur tersebut digunakan untuk mendeteksi *request* dari *browser client*, kemudian server akan memberikan catatan sebelum memberikan respons kembali ke client. Berikut contoh server yang mendukung patch menggunakan dua format dokumen secara hipotesis.

```
[request]

   OPTIONS /example/buddies.xml HTTP/1.1
   Host: www.example.com

[response]

   HTTP/1.1 200 OK
   Allow: GET, PUT, POST, OPTIONS, HEAD, DELETE, PATCH
   Accept-Patch: application/example, text/example
```

Gambar 2.4 Contoh catatan patch pada server.

2.3 Packet Analyzer

Packet Analyzer atau biasa disebut juga *packet sniffer*, Analisis jaringan, atau Analisis protokol, adalah sebuah program komputasi yang digunakan untuk mengetuk jaringan dan membaca data biner yang mengalir. Data biner tidak dapat dibaca oleh manusia. Oleh karena itu, analisis protokol harus dilakukan dan data harus dibedah menjadi sebuah paket dan bidang yang terpisah. Kemudian header dan paket diterjemahkan dan ditafsirkan yang sesuai, proses ini sering disebut sebagai paket *sniffing*, yang cenderung menjadi salah satu istilah yang paling populer digunakan pada saat ini (Orebaugh & Ramirez, 2004; Sanders, 2011).

Proses paket *sniffing* akan dibagi menjadi 3 langkah, yaitu:

a. Capturing, yang akan digunakan paket sniffer.
b. Decoding, hasil capture binary data yang telah dikonversikan menjadi bentuk yang dapat dibaca, agar pengguna mudah memahami.
c. Analyzing, langkah terakhir melibatkan analisis yang sebenarnya data yang diambil dan diterjemahkan. Pada langkah ini, protokol yang digunakan akan diinterpretasikan dari informasi yang diekstrak dari lalu-lintas jaringan. Setelah diverifikasi, data dianalisis sesuai dengan spesifikasi protokol, menghasilkan data terstruktur dari paket yang terpisah.

2.4 Tipikal Penggunaan

Tipikal penggunaan pada paket sniffing dan Analisis jaringan terbagi menjadi dua tipikal/kategori, yaitu

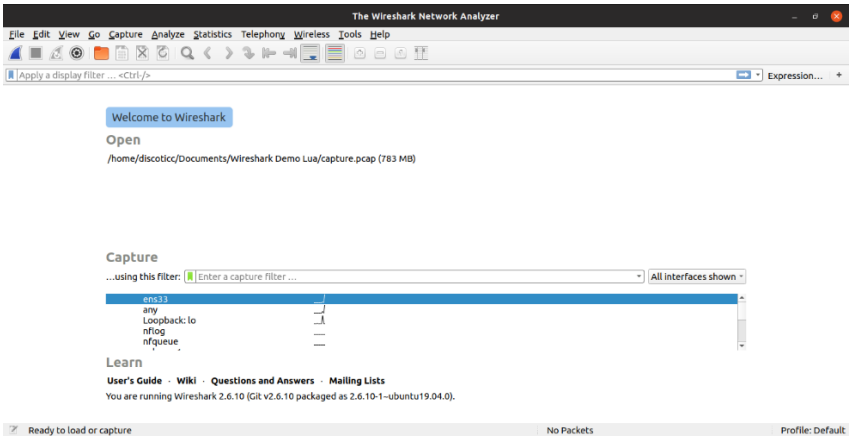
a. Benevolent user: Pengguna yang menggunakannya sebagai pelindung, uji coba, dan menjadi administrasi jaringan yang memantau di lingkungan lokal.
b. Malicious user: Pengguna yang menggunakannya untuk mendengar percakapan pengguna lain *Man-in-the-Middle* untuk mendapatkan informasi, menyerang, dan eksploitasi jaringan ke pengguna lain.

Selain penggunaan, analisa paket dan sniffer digunakan oleh peretas untuk melakukan tindakan jahat. Dengan alat bantu sniffer, alat untuk mengintip percakapan pribadi menjadi lebih mudah. Biasanya tujuan peretas adalah mengambil informasi sensitif saja dari pengguna jaringan atau perangkat komputer yang ditargetkan contohnya adalah nama pengguna, kata sandi, nomor kartu debit atau kredit, *cookie* dan *session ID.* Enkripsi adalah alat keamanan yang diperlukan untuk melindungi informasi dalam bentuk acak, tetapi masih banyak situs web mengirim dan menerima data dalam bentuk teks biasa yang membuatnya mudah dilihat dan didengar (eavedrop) dengan sniffing *traffic* jaringan. Jika protokol kriptografi digunakan, maka data akan dilakukan enkripsi (Chomsiri, 2007, 2008; Fuentes & Kar, 2005; S. Ansari dkk., 2002).

2.5 Wireshark

Wireshark merupakan salah satu dari sekian banyak tools penganalisis jaringan yang banyak digunakan oleh administrasi jaringan untuk menganalisis jaringan termasuk beberapa protokol didalamnya. Wireshark banyak diminati pengguna karena tampilan antarmuka secara

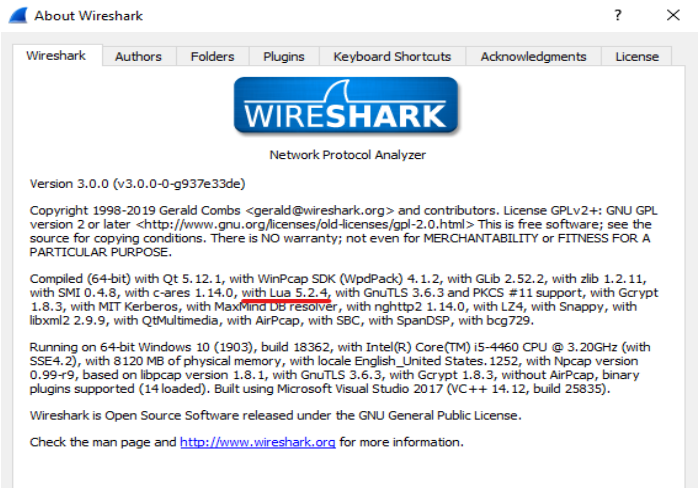
grafis (Graphical User Interface). Fitur utama Wireshark adalah mampu menangkap paket – paket data atau informasi melalui jaringan. Semua jenis paket data dalam berbagai format protokol mudah ditangkap dan dianalisis. Oleh karena itu, tools ini dapat dipakai untuk *sniffing* dengan menangkap paket – paket yang melewati dan menganalisis (Widodo, 2012).



Gambar 2.5 Graphical User Interface Wireshark.

2.6 Lua Scripting

Lua adalah sebuah pemrograman yang ringan dan multi-paradigma dirancang untuk penggunaan yang disimpan di dalam aplikasi. Di bidang jaringan komputer, Lua adalah sebuah *cross-platform*, karena ditulis dalam ANSI bahasa C, dan memiliki C API yang sederhana. Lua dapat digunakan untuk menulis beberapa dissector, post-dissector, dan tap. Meskipun kemungkinan untuk menulis dissector di bahasa Lua, seperti di aplikasi Wireshark pada umumnya ditulis dalam bahasa C karena bahasa C lebih cepat daripada Lua, tetapi Lua diketik secara dinamis, ditafsirkan dari *opcodes* dan memiliki manajemen memori secara otomatis dengan pengumpulan uraian menjadikannya ideal untuk konfigurasi, *scripting*, serta uji coba yang cepat. Scripting Lua tersedia di Wireshark seperti Gambar 2.6 berikut. (Ierusalimschy dkk., 2003).



Gambar 2.6 Wireshark mendukung pemrograman bahasa Lua sebagai plug-in.

Lua pada Wireshark mempunyai kelebihan dan kekurangan, kelebihan dari Lua adalah sebagai berikut (Bjørlykke, 2009):

a. Membuat prototipe dengan mudah, implementasi dan uji coba.
b. Dibutuhkan sedikitnya souce code.
c. Tidak ada manajemen penyimpanan.
d. Mudah untuk berbagi dengan yang lain.

Sedangkan kelemahan dari Lua adalah sebagai berikut:

a. Kinerja lebih lambar dari bahasa/*scripting* C.
b. Hanya sebagian fungsi *dissector*.
c. Tidak semua code didistribusikan oleh Wireshark.
d. Belum digunakan oleh banyak pengguna.

2.7 Ettercap

Ettercap adalah sebuah aplikasi paket yang komprehensif untuk serangan *man-in-the-middle* pada LAN, karena kemampuan spoofing pada ARP. Aplikasi ini berbasis linux, gratis, open source, dan lintas platform. Tiga antarmuka berbeda ditawarkan kepada pengguna yaitu terminal, GUI, dan ncurses1.

Ettercap dipilih untuk mengidentifikasi ke entitas jaringan untuk memberikan informasi tentang alamat IP host, MAC Address, dan penggunaan port. Jika dibandingkan dengan aplikasi sniffing lainnya, ettercap bekerja lebih baik dari semuanya (Vollmer & Manic, 2014). Ettercap memiliki banyak fitur, salah satu dari fitur ini adalah *sniffing* koneksi secara langsung, melakukan analisis jaringan ke host, dan menyaring konten dengan cepat.