# BAB I
# PENDAHULUAN

## 1.1. LATAR BELAKANG

Autentikasi WEB merupakan proses pengenalan pengguna yang bertujuan melindungi akses sistem informasi dari pengguna diluar hak atau menentukan hak akses pengguna yang diberi batasan tertentu. Autentifikasi pada WEB menggunakan *protocol* HTTP POST/GET untuk melakukan pertukaran data dari *client* WEB ke *server*. *protocol* tersebut mengirimkan data berupa *plain text* dari *client* ke *server*. Proses pengiriman pada *protocol* HTTP ini menjadi tidak aman saat *protocol* yang melakukan resolusi atau ARP memiliki celah yang disebut dengan ARP *spoofing* yang kemudian mengerucut pada metode peretasan yaitu *MAN in THE middle ATTACK* (MITM). MITM adalah proses dimana meyakinkan komunikasi yang terjadi antara dua *host* bahwa komputer di tengah (*attacker*) merupakan *host* yang dituju . Hal ini dapat dicapai dengan *spoofing* nama *domain* jika sistem menggunakan DNS untuk mengidentifikasi *protocol host* lain atau bisa disebut dengan ARP *Spoofing* dalam lan.

*Protocol* HTTP memiliki versi *secure* yang disebut HTTPS. HTTPS menyandikan data sesi menggunakan *protocol* SSL (*Secure Socket layer*) atau *protocol* TLS (*Transport Layer Security*). Kedua *protocol* tersebut memberikan *perlindungan* yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks*. Tingkat keamanan tergantung pada ketepatan dalam mengimplementasikan pada *browser* WEB dan perangkat lunak *server* dan didukung oleh algoritma penyandian yang aktual. *Protocol* transmisi ini relatif lebih aman dan dapat menjaga integritas data yang diakses pengguna dari penyadapan.

Penerapan *protocol* https memiliki permasalahan saat diterapkan pada autentikasi *firewall* atau sering disebut *captive portal server* yang mengautentikasi calon pengguna layanan akses internet berbasis web/http. Pada saat autentikasi client dilakukan pada *protocol* https, *browser client* yang berfungsi menampilkan *interface* halaman *login* akan menampilkan *certificate error* meskipun certificate

SSL *server captive portal* telah terdaftar pada penyedia layanan otorisasi *certificate* SSL. *Client* tetap dapat melakukan proses autentikasi dengan mengabaikan peringatan *certificate error* yang artinya *client* mengabaikan validasi sertifikat SSL. *Error certificate* baru akan hilang setelah client diautentikasi oleh *server captive portal*.

Hal lain yang terjadi adalah autentikasi *captive portal hotspot* tidak lagi menggunakan https. Penggunaan https pada autentikasi captive portal dirasa akan merepotkan client. Permasalahan tersebutlah yang akan membuat seorang teknisi memikirkan penggunaan http dibanding https.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan dapat ditarik perumusan masalah yaitu :

1) Apakah client yang mengakses halaman login captive portal tetap aman dari serangan *password sniffing* (*eavesdropping*) saat mengabaikan peringatan error sertifikat SSL pada browser.
2) Bagaimanakah cara agar *client* tetap aman dari *password sniffing* (*eavesdropping*) saat melakukan proses *login captive portal* jika *client* lebih memilih menggunakan *protocol* http ketimbang https.
3) Jika autentikasi pada http telah aman, apakah penggunaan http dengan pengamanan tertentu tersebut dapat menggantikan autentikasi menggunakan https.

## 1.3. Batasan Masalah

MITM biasanya dilakukan pada jaringan lokal (LAN). Jadi, akan digunakan 3 buah *computer*/*host*, yaitu *client host*, *attacker*, dan *host server* dengan captive portal. Sedangkan, captive portal *server* yang meng-autentikasi client dengan web/*protocol* http akan diuji keamanannya menggunakan MITM *Password sniffing* (eavesdropping) dengan menerapkan *protocol* ssl pada autentikasi *client*.

Pada autentikasi web/http captive portal internet *hotspot* dengan ssl akan selalu menampilkan *error* pada *browser*. Meskipun sertifikat browser telah terdaftar pada otorisasi penyedia layanan sertifikat *ssl*.

## 1.4. Tujuan

1) Menguji keamanan protocol HTTPS/SSL dari serangan MITM *password sniffing* saat client mengabaikan peringatan *certificate error* pada prosess autentikasi *captive portal*.
2) Mengamankan autentikasi http pada *captive portal server*.
3) Membandingkan keuntungan dan kerugian autentikasi menggunakan https dengan autentikasi http yang telah ditambah dengan pengamanan tertentu.

## 1.5. Manfaat

1) Mengetahui keamanan penggunaan protocol SSL saat client yang melakukan autentikasi mengabaikan peringatan *error certificate* SSL dari serangan MITM password sniffing (eavesdropping).
2) Memungkinkan captive portal meng-autentikasi *client* dengan menggunakan *protocol* http yang tidak akan menampilkan *certificate* error namun dengan tidak mengabaikan keamanan data *username* dan *password client* dari serangan MITM *password sniffing* (*eavesdropping*).
3) Mengetahui kondisi yang cocok penggunaan protocol https atau http dengan pengamanan tertentu dengan mengamati keuntungan dan kerugian penerapannya.