Implementasi SSL untuk Keamanan Data pada Sistem Inventaris (Studi Kasus: Sanggar Tari Natya Lakshita)

Sadr Lufti Mufreni*, Danur Wijayanto, Sri Maryani*****
* ** *** Program Studi Teknologi Informasi, Universitas 'Aisyiyah Yogyakarta
* sadr@unisayogya.ac.id, **danurwijayanto@unisayogya.ac.id, *** srimaryani9104@gmail.com

ABSTRACT

The issue of data security and confidentiality of communications on the network is one of the most important aspects. Data security in the Inventory System is currently not completely safe; this is because the Inventory System has not implemented client and server communication security. Secure Socket Layer (SSL) is one of the efforts that can be made to prevent illegal attacks. This study applied the SSL method, in which SSL acted as a data security protocol in client and server communication networks. The purpose of this research was to implement SSL in order to improve data security in the Inventory System at Sanggar Tari Natya Lakshita. The process of observing and analyzing the data security of the Inventory System was carried out using Wireshark tools. The results showed that the implementation of SSL in the Inventory System was successful; this was evidenced by the results of observing and analyzing the data packets captured by Wireshark; the data was not easy to read. The success rate of doing 3 trials, namely logging in, inputting and deleting the data packet, was 100% encrypted.

Keyword: *SSL Implementation, Data Security, Wireshark*

1. Pendahuluan

Perkembangan teknologi informasi saat ini sangat pesat merambah ke berbagai bidang khususnya di bidang jaringan komputer dan berpengaruh dikehidupan sehari-hari bagi pihak yang memanfaatkannya, hal ini disebabkan karena teknologi informasi memudahkan masyarakat dalam mendapatkan suatu informasi pada jaringan *internet*. Instansi yang memanfaatkan adanya teknologi informasi adalah Sanggar Tari Natya Lakshita [1]. Sanggar Tari Natya Lakshita merupakan sebuah lembaga yang bergerak di bidang pendidikan dan pelatihan yang memikul tanggung jawab yang besar untuk pelestarian dan pengembangan tari dengan tidak meninggalkan kaidah-kaidah yang sudah ada[2].

Teknologi yang digunakan oleh Sanggar Tari Natya Lakshita adalah Sistem Inventaris. Sistem Inventaris merupakan sistem yang mencatat semua data koleksi yang bertujuan untuk mempermudah penyimpanan dan pencarian data koleksi di Sanggar Tari Natya Lakshita, sistem ini berbasis *web* dengan *Framework Laravel* untuk mempermudah pihak sanggar dalam menjalankan proses bisnis [3].

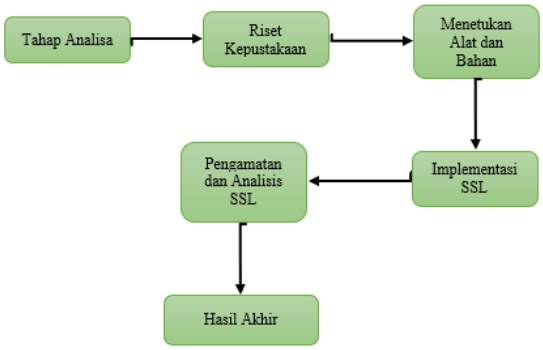
Masalah keamanan data dan kerahasiaan Sistem Inventaris adalah salah satu aspek yang paling penting dari suatu data atau informasi, mengingat luasnya jangkauan *internet* dari sisi teknologi maupun sisi regional wilayah [4]. Keamanan yang ada pada Sistem Inventaris saat ini dapat dikatakan belum benar-benar aman, hal ini dikarenakan Sistem Inventaris belum menerapkan keamanan komunikasi antara *client* dan *server*, untuk memenuhi kebutuhan keamanan dalam komunikasi data salah satu solusi yang ditawarkan kepada Sanggar Tari Natya Lakshita adalah dengan mengimplementasikan *Secure Socket Layer* (SSL) pada Sistem Inventaris.

SSL adalah suatu protokol yang dapat digunakan dalam komunikasi dan transaksi yang cukup mumpuni. SSL bertindak sebagai protokol yang mengamankan komunikasi antara *client* dan *server*[5]. Protokol ini memberikan fasilitasi penggunaan enkripsi untuk data yang rahasia dan membantu menjamin integritas informasi yang dipertukarkan antara *website* dan *web browser* [6].

Penelitian ini bertujuan untuk mengimplementasikan SSL pada Sistem Inventaris yang diharapkan dapat meningkatkan keamanan data Sistem Inventaris pada Sanggar Tari Natya Lakshita dan meningkatkan kinerja sistem menjadi lebih optimal serta agar dapat mencegah terjadinya serangan [7]. Berbeda dengan penelitian yang telah dilakukan sebelumnya [8], [9], penelitian ini menggunakan Sistem Operasi Windows dan menjelaskan secara rinci mengenai implementasi SSL. Pengamatan dan analisis pada penelitian ini menggunakan *Software Wireshark* dengan menghasilkan sebuah laporan terperinci mengenai keamanan sebelum dan sesudah mengimplementasikan SSL pada Sistem Inventaris[10].

2. Tahapan Penelitian

Tahapan yang dilakukan dalam penyelesaian penelitian ini melakukan tahap Analisa, riset kepustakaan, tahap menentukan alat dan bahan, tahap implementasi SSL, tahap pengamatan dan analisis SSL, hasil akhir. Tahapan penelitian dapat dilihat pada Gambar 2.1**.**



Gambar 2.1 Tahapan Penelitian

2.1. Tahap Analisa

Tahapan ini dilakukan dengan tujuan untuk mendapatkan informasi yang ada pada Sanggar Tari Natya Lakshita terkait Sistem Inventaris yang digunakan. Informasi tersebut di peroleh dengan melakukan observasi dan wawancara untuk keakuratan data dan keaslian informasi yang ada[11].

2.2. Tahap Riset Kepustakaan

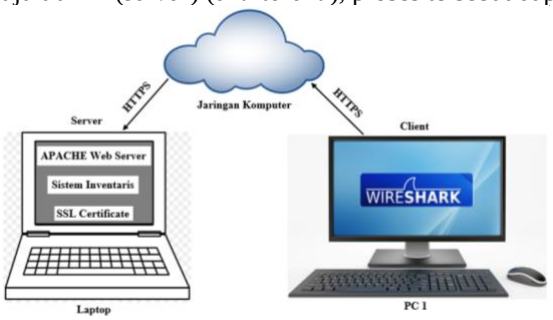
Tahapan ini dilakukan dengan tujuan untuk mencari dan memperoleh informasi yang di perlukan sebagai bahan referensi terkait kasus dan digunakan sebagai bahan acuan saat penelitian, studi *literatur* menjelaskan dasar teori atau sebuah kajian yang di dapatkan dengan membaca jurnal, artikel ataupun sumber lainnya.

2.3. Tahap Menentukan Alat dan Bahan

Tahapan ini melakukan penentuan alat dan bahan yang akan digunakan pada saat penelitian berlangsung, kebutuhan yang digunakan dibagi menjadi 2 bagian meliputi perangkat keras dan perangkat lunak. Perangkat keras meliputi Laptop HP dengan Sistem Operasi Windows 10 sebagai *server* dan Komputer Lab sebagai *client* dan perangkat lunak yang digunakan adalah *Wireshark*.

2.4. Tahap Implementasi SSL pada Sistem Inventaris

SSL akan dipasang pada Sistem Inventaris, proses implementasi SSL memiliki beberapa tahapan. SSL dapat merubah alamat Sistem Inventaris dari alamat HTTP menjadi HTTPS. SSL diterapkan untuk memastikan keamanan dalam bertransaksi di internet antara *web server* dan *browser* dari *client*. Keamanan SSL hanya berlaku dari titik A (*client*) menuju titik B (*server*) (*end-to-end*), proses tersebut dapat dilihat pada gambar 2.2



Gambar 2.2 Topologi SSL

2.5. Tahap Pengamatan dan Analisis

Tahapan ini melakukan pengamatan dan analisis SSL menggunakan *Wireshark*. Pengamatan dan analisis ini dilakukan pada Sistem Inventaris berbasis *web* yang ada pada Sanggar Tari Natya Lakshita sebelum dan sesudah dipasang SSL. *Wireshark* meng-*capture* lalu lintas paket saat meload *web* tersebut[12].

2.6. Tahap Hasil Pengamatan dan Analisis

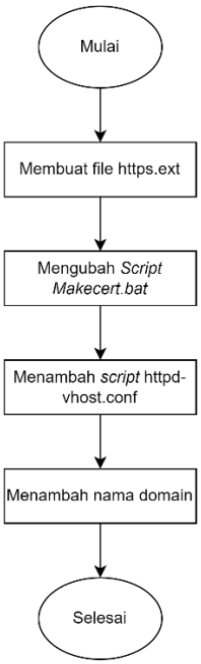
Tahapan ini berisi penjelasan terkait hasil yang diperoleh setelah dilakukan pengataman dan analisis SSL pada Sistem Inventaris, pengamatan dan analisis SSL menggunakan *Wireshark*.

3. Hasil dan Pembahasan

Hasil dan pembahasan merupakan tahap pemaparan hasil yang diperoleh dari proses implementasi SSL, analisis sebelum dan sesudah melakukan implementasi SSL dengan menggunakan *Wireshark*.

3.1. Tahap Implementasi SSL

SSL akan dipasang pada Sistem Inventaris yang sudah terpasang di dalam *server*. Proses implementasi SSL disesuaikan dengan kondisi pada pengembangan Sistem Inventaris di Sanggar Tari Natya Lakshita, proses setiap tahapan implementasi SSL pada Sistem Inventaris dapat dilihat pada gambar *flowchart* 3.1.



Gambar 3.1 *Flowchart* Implementasi SSL

3.1.1. Membuat *File* HTTPS.ext

File https.ext perlu ditambahkan "`subjectAltName`" dengan nilai kunci "`@alt_names`" di dalamnya terdapat daftar lengkap nama DNS dan alamat *IP Address* yang akan diproses oleh sertifikat *server* pada saat menvalidasi permintaan *client*. *File* https.ext dibuat lalu disimpan pada *folder* Xampp yang ada pada *drive* C:\xampp\apche, *source code* pada *file* https.ext dapat dilihat pada *Source code* 3.1

Source code 3.1 *file* https.ext

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature,
nonRepudiation, keyEncipherment,
dataEncipherment
subjectAltName = @alt names

[alt_names]
DNS.1 = localhost
DNS.2 = didik_nini_thowok.com
```

3.1.2. Mengubah *Source Code* pada *File Makecert*

Makecert digunakan untuk membuat sertifikat, bukan sebagai mekanisme validasi. Artinya, sertifikat apapun yang telah dibuat tidak akan terpengaruh meskipun *makecert* tidak lagi tersedia ataupun telah dihapus. perubahan *source code* pada *file makecert* dapat dilihat pada *Source code* 3.2

Sorce code 3.2 *file makecert*

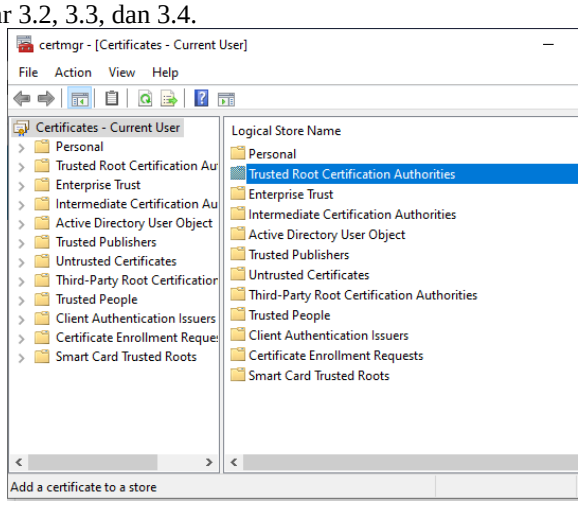
Sebelum	`bin\openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 365`
Sesudah	`bin\openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 500 -sha256 -extfile https.ext`

Web yang menggunakan SSL memiliki sertifikat x509 di *server web* dan menggunakannya untuk mengenkripsi dan mendeskripsi data dengan cepat. Sertifikat berlaku untuk 500 hari kedepan atau 1 tahun 4 bulan 15 hari setelah penandatanganan, tanggal mulai diatur ke waktu saat ini dan tanggal akhir berlakunya sertifikat. Perubahan pada *source code* disimpan, selanjutnya membuka *command prompt* kemudian ketik "`makecert`" maka akan muncul suatu perintah yang harus diikuti, Kemudian kita akan diminta untuk mengisi berberapa hal yang berkaitan dengan domain dan lokasi tempat kita berada, seperti ditunjukkan pada Tabel 3.1.

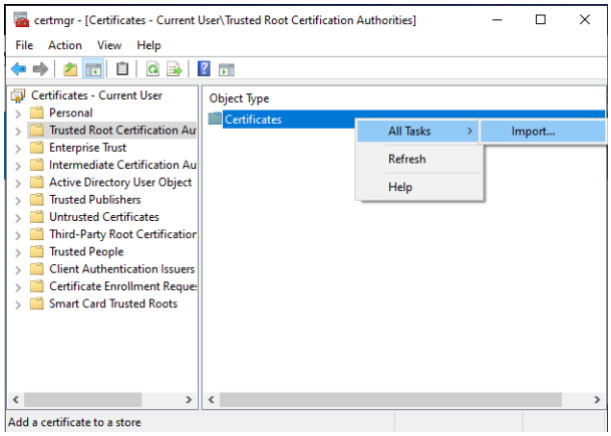
Tabel 3.1 Perintah *Makecert*

Perintah *Makecert*
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
A challenge password []:
An optional company name []:
Enter pass phrase for privkey.pem:.
Press any key to continue…

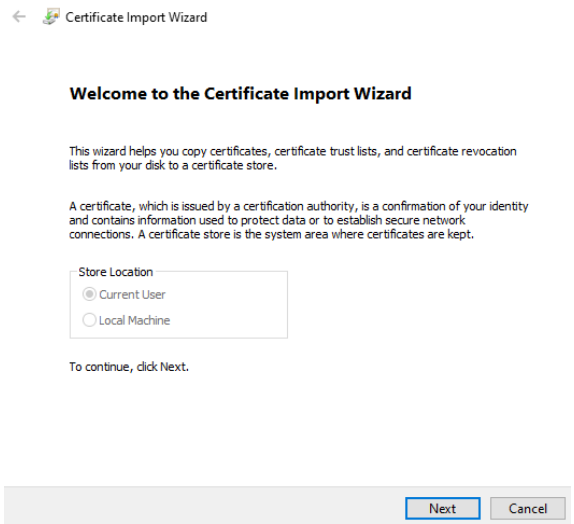
Setelah mengikuti perintah yang muncul pada *command prompt*, penginstalan SSL sudah selesai. Membuka *manage user certificates* di *search* Windows ketikkan *certmgr.msc* lalu enter. *Manage User Certificates* akan muncul pada layar, pilih *trusted root certification authorities* dan klik kanan pada *certificates* pilih *all tasks* lalu *import*. Tampilan *Manage User Certificates* dan tampilan *trusted root certification authorities* seperti pada Gambar 3.2, 3.3, dan 3.4.



Gambar 3.2 *Manage User Certificates*

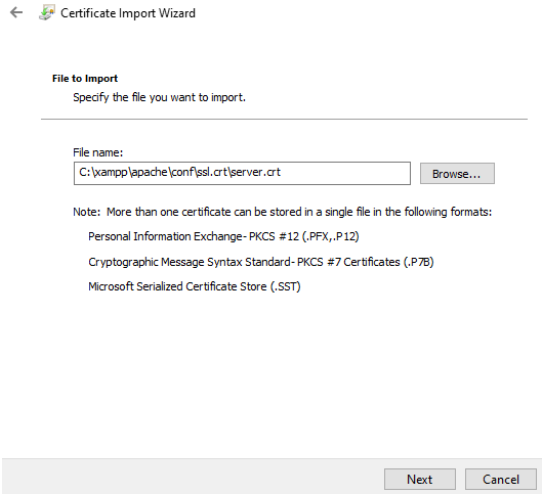


Gambar 3.3 Tampilan *Trusted Root Certification Authorities*



Gambar 3.4 *Certificates Import Wizard*

Tampilan *certificates import wizard* akan muncul pada layar. SSL akan disimpan di dalam directori C:\xampp\apache\conf\ssl.crt\server.crt. Tampilan direktori tempat akan disimpannya sertifikat SSL dapat dilihat pada Gambar 3.5



Gambar 3.5 *Direktory* SSL

3.1.3. Menambah File httpd-vhosts.conf di XAMPP

Virtualhost ini dapat membuat lebih dari satu *website* dengan konten yang berbeda dalam sebuah *server*. *Source code* konfigurasi pada *virtualhost* akan ditambahkan ke dalam *file* httpd-vhosts.conf, lalu akan disimpan kembali. *Restart apache* setelah proses *edit* selesai. *Source code* konfigurasi *virtualhost* dapat dilihat pada *Source code* 3.3

Source code 3.3 konfigurasi *virtualhost*

```
<VirtualHost 127.0.0.1:80>
    DocumentRoot "C:/xampp/htdocs/Develompent/didik nini thowok/public"
    ServerName didik nini thowok.com
    ServerAlias www. didik nini thowok.com
</VirtualHost>

<VirtualHost 127.0.0.1:443>
    DocumentRoot "C:/xampp/htdocs/Develompent/
didik_nini_thowok/public"
    ServerName didik_nini_thowok.com
    ServerAlias www. didik_nini_thowok.com
    SSLEngine on
    SSLCertificateFile "C:/xampp/apache/conf/ssl.crt/server.crt"
    SSLCertificateKeyFile "C:/xampp/apache/conf/ssl.key/server.key"
    <Directory "C:/xampp/htdocs/Develompent/ didik nini thowok/public">
        Options All
          AllowOverride All
          Require all granted
    </Directory>
</VirtualHost>
```

3.1.4 Menambah Nama Domain pada File Hosts

Nama domain ditambahkan ke dalam *file host Windows*. *File hosts* terdapat di dalam direktori "C:/WINDOWS/system32/drivers/etc/hosts. klik 2 kali *file hosts* untuk membuka, setelah *file hosts* terbuka tambahkan nama *host* baru pada bagian akhir, *source code file hosts* yang dapat dilihat pada *Source code* 3.4

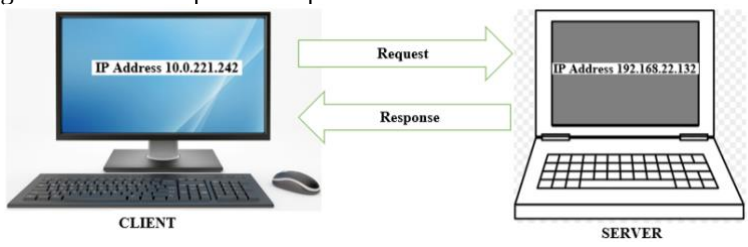
Source code 3.4 *file hosts*

```
127.0.0.1       didik_nini_thowok.com
```

Penambahan *host* baru dengan nama domain dan *IP localhost*. Nama domain yang dibuat adalah didik_nini_thowok.com dengan *IP localhost* 127.0.0.1, setelah selesai melakukan *editing file hosts* kemudian simpan *file* lalu *restart apache*.

3.2 Konfigurasi *Server* dan *Client*

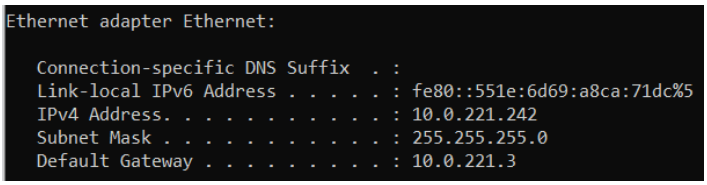
Konfigurasi *server* dan *client* dilakukan dengan tujuan agar Sistem Inventaris yang terinstall pada *server* dapat diakses oleh *client*[13]. *Server* dan *client* merupakan konektifitas pada jaringan yang membedakan fungsi pada sebuah komputer apakah sebagai *server* atau *client*, *server* bertugas memberikan pelayanan maupun informasi kepada *client* sedangkan *client* menerima fasilitas pelayanan maupun informasi yang disediakan oleh *server* [11]. Topologi *client/server* dapat dilihat pada Gambar 3.6



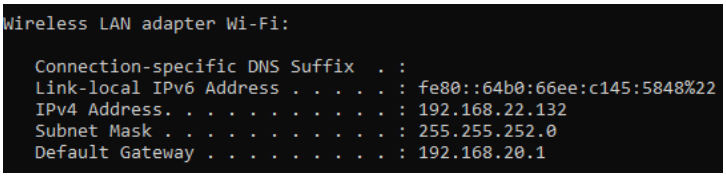
Gambar 3.6 Topologi *client/server*

Tes *ping* dilakukan untuk mengetahui komputer *server* dan komputer *client* sudah terhubung atau belum, pengecekan *IP Address* dari kedua komputer dilakukan terlebih dahulu sebelum melakukan tes *"ping"*. pengecekan *IP Address* dilakukan dengan mengetik "`ipconfig`" pada *command Prompt* kemudian enter, maka akan muncul *IP Address* komputer *client* memiliki *IP Address* 10.0.221.242 *subnet maks* 255.255.255.0 dan *Default Gateway* 10.0.221.3, komputer *server* memiliki *IP Address* 192.168.22.132 *Subnet Mask*

255.255.255.0 dan *default gateway* 192.168.20.1. *IP Address* komputer *client* dan *server* dapat dilihat pada Gambar 3.7 dan Gambar 3.8.

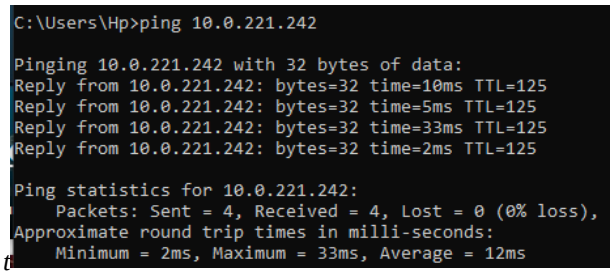


Gambar 3.7 *IP Address Client*

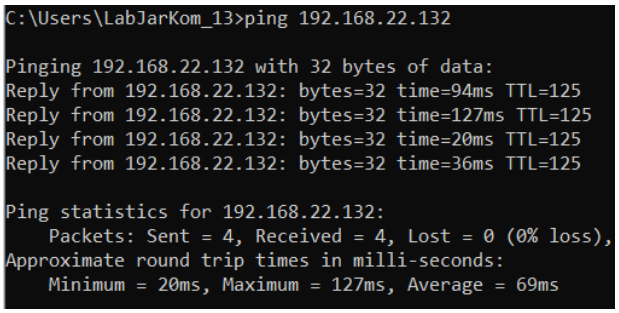


Gambar 3.8 *IP Address Server*

Tes "PING" dapat dilakukan dari komputer *server* ke komputer *client*. Tujuan *ping* untuk menentukan apakah sistem apa alamat IP tertentu ada dan saat ini berfungsi, jalur ke sistem tersebut dapat ditemukan. *Server* melakukan *ping* pada *client* dengan cara ketik "`ping 10.0.221.242`" pada *command prompt* lalu enter, kemudian *client* merespon dengan *reply from* 10.0.221.242. Hasil tes "PING" dari *server* ke *client* dan tes *"ping"* dari *client* ke *server* dapat dilihat pada Gambar 3.9 dan 3.10.



Gambar 3.9 Tes *ping server* ke *client*

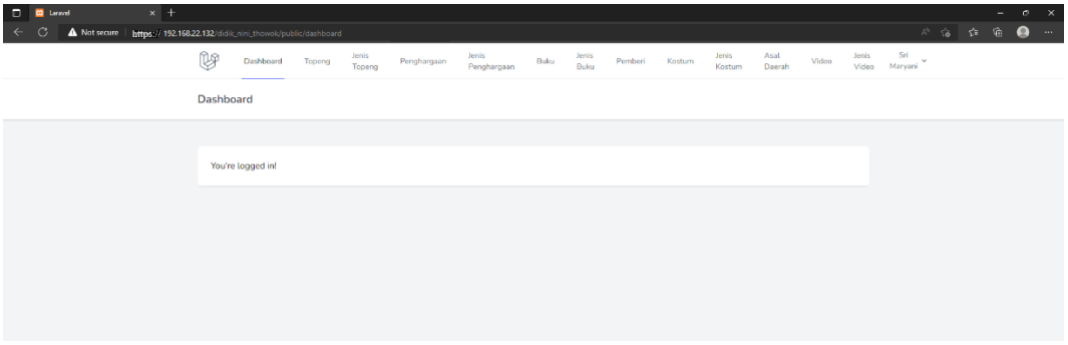


Gambar 3.10 Tes *ping client* ke *server*

Hasil *ping* dari *client* ke *server*. *Client* melakukan *ping* ke *server* dengan cara ketik "`ping 192.168.22.132`" pada *command prompt* lalu enter, kemudian *server* merespon dengan *reply from* 192.168.22.132 dengan `pinging 192.168.22.132 with 32 bytes of data` artinya telah dilakukan *ping* ke alamat *IP* tersebut dengan 32 *bytes* data, paket yang dikirim tidak ada yang hilang (dikirim 4 diterima 4). Respon yang ada dari kedua komputer tersebut membuktikan bahwa *server* sudah terhubung dengan *client* [14]. Mengatur Sistem Inventaris agar dapat diakses oleh komputer *client* dengan menghilangkan tanda # pada bagian `LoadModule vhost_alias_module modules/mod_vhost_alias.so` dalam s*ource code* pada *file* httpd.conf.

Komputer *client* mengakses Sistem Inventaris dengan mengetik alamat *url* *https://192.168.22.132/didik_ninik_thowok/public/login*. Sistem Inventaris dapat diakses oleh komputer *client* ketika posisi Xampp yang ada pada komputer *server* dalam keadaan aktif, tampilan Sistem Inventaris yang

diakses pada komputer *client* dapat dilihat pada Gambar 3.11.

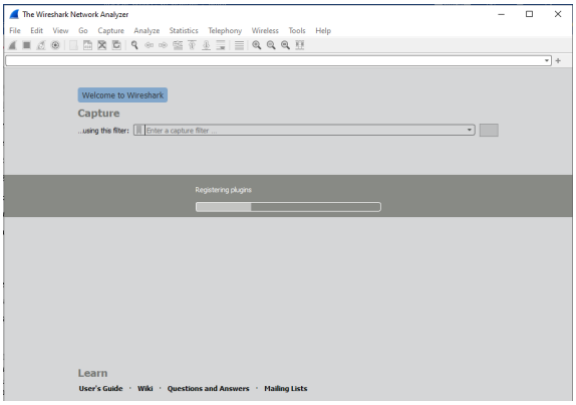


Gambar 3.11 Tampilan Sistem Inventaris

IP Address pada komputer *server* dapat berubah sewaktu-waktu hal ini dikarenakan *IP Address* yang digunakan adalah *IP Address Dynamic*. *IP Address Dynamic* merupakan IP Address yang diberikan kepada komputer secara otomatis dan akan selalu berubah-ubah atau tidak tetap setiap saat di dalam jaringan *private* maupun publik [15].

3.3 Tahap Pengamatan dan Analisis

Pengamatan dan analisis Sistem Inventaris sebelum dan sesudah dipasang SSL dengan *Wireshark*, proses *capture* ini dilakukan pada komputer *client*. Peneliti sebelumnya telah melakukan instalasi *Wireshark* pada komputer *client*. *Wireshark* diinstall di dalam komputer *client*, dimana komputer *client* akan mengakses Sistem Inventaris untuk melakukan uji coba *capture* paket data dengan *Wireshark*. Tampilan *Wireshark* dapat dilihat pada Gambar 3.12



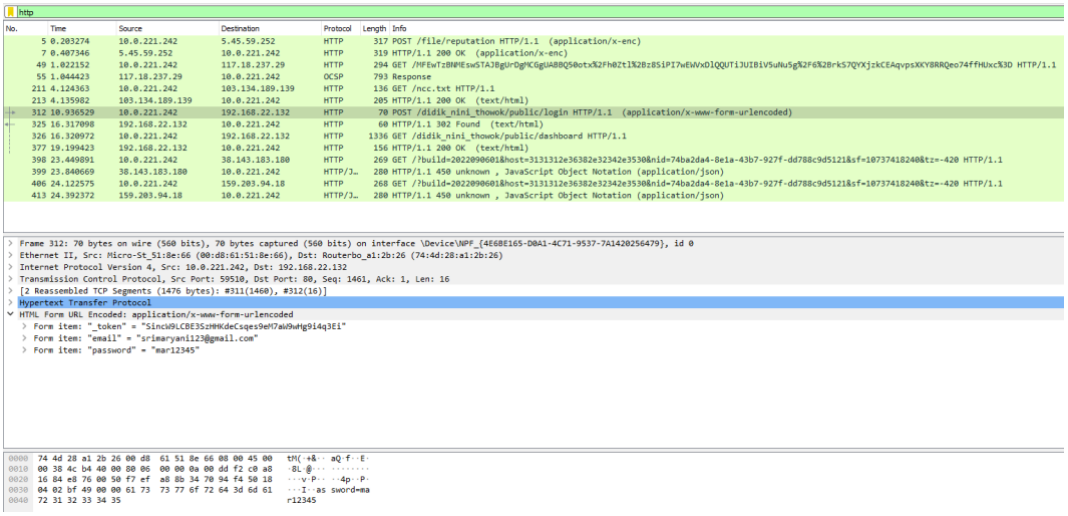
Gambar 3.12 Tampilan awal *Wireshark*

3.3.1 Hasil Sebelum Implementasi SSL

Hasil pengamatan dan analisa dengan *Wireshark* dengan *capture* sistem inventaris sebelum SSL dipasang. Pengamatan dan analisis dengan *Wireshark* bertujuan untuk mengetahui perbedaan yang ada sebelum dan sesudah SSL dipasang pada Sistem Inventaris.

Percobaan Login Sistem Inventaris

Hasil *capture Ethernet* akan muncul paket data yang tertangkap oleh *Wireshark* yaitu paket data *login* pada Sistem Inventaris. Daftar paket data yang tertangkap dilengkapi dengan nomor paket, *Time*, *IP Address* asal, *IP Address* tujuan, *protocol* yang digunakan yaitu HTTP, serta panjang paket tersebut berapa *bytes*. Informasi detail data *Frame* 312 dengan panjang 70 *bytes*. Tanggal paket data berlangsung pada 14 September 2022 jam 11:54:22 dengan protokol HTTP dan menggunakan *port* 80.

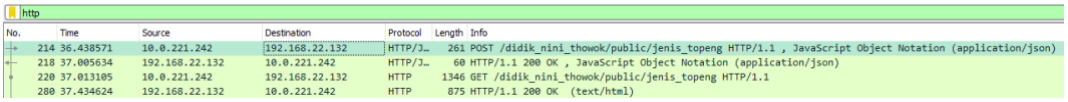


Gambar 3.12 *Capture* HTTP

Detail informasi yang diperoleh yaitu terdapat *request method*: POST (mengambil dari inputan user), *request URL*: /didik_nini_thowok/public/login, *request version*: HTTP/1.1, *Host*: 192.168.22.132\r\n. *Protocol* yang digunakan adalah *protocol* http, maka *Wireshark* dapat menangkap detail paket data dengan informasi *"_token"*, e-mail dan *password*. Informasi tersebut muncul dikarenakan paket data yang tertangkap oleh *Wireshark* pada saat *user* sedang melakukan *login* dan *user* memberikan informasi seperti *input* data seperti e-mail dan *password*.

Percobaan Input Data dan Delete Data

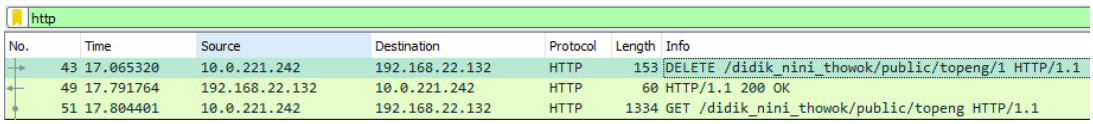
Hasil *capture Ethernet*, muncul paket data yang tertangkap oleh *Wireshark* yaitu paket data *input* data jenis topeng pada Sistem Inventaris. Informasi yang di peroleh sama dengan informasi sebelumnya yang membedakan hanya prosesnya yaitu *POST login* dan *POST input* data jenis topeng. Detail informasi dapat dilihat pada Gambar 3.13



Gambar 3.13 *Input* Data Jenis Topeng

Informasi detail data *Frame* 214 dengan panjang 261 *bytes*. Tanggal paket data berlangsung pada 14 September 2022 jam 12:21:20 dengan *protocol* HTTP dan menggunakan *port* 80. Informasi detail data *Ethernet* Dan *Internet Protocol Version 4*. Informasi detail *ethernet II* terdapat *MAC* dari *source* dan *MAC* dari *destination*, dimana *MAC* dari *source* yaitu `Micro-St_51:8e:66 (00:d8:61:51:8e:66)` dan *MAC* dari *destination* yaitu `Routerbo_a1:2b:26 (74:4d:28:a1:2b:26)`. Detail informasi pada *Internet Protocol Version* 4 terdapat *IP Address* asal dan *IP Address* tujuan, dimana *IP Address* asal yaitu 10.0.221.242 dan *IP Address* tujuan 192.168.22.132. Informasi detail terkait *input* data jenis topeng, protokol yang digunakan adalah *protocol* http, *Wireshark* juga dapat menangkap detail paket data dengan informasi *member*: ID dan Nama.

Hasil *capture Ethernet*, muncul paket data yang tertangkap oleh *Wireshark* yaitu paket data *delete* data topeng pada Sistem Inventaris. Tampilan Hasil *capture Ethernet*, muncul paket data yang tertangkap oleh *Wireshark* yaitu paket data *delete* Topeng pada Sistem Inventaris dapat dilihat pada Gambar 3.14



Gambar 3.14 *Delete* Data Topeng

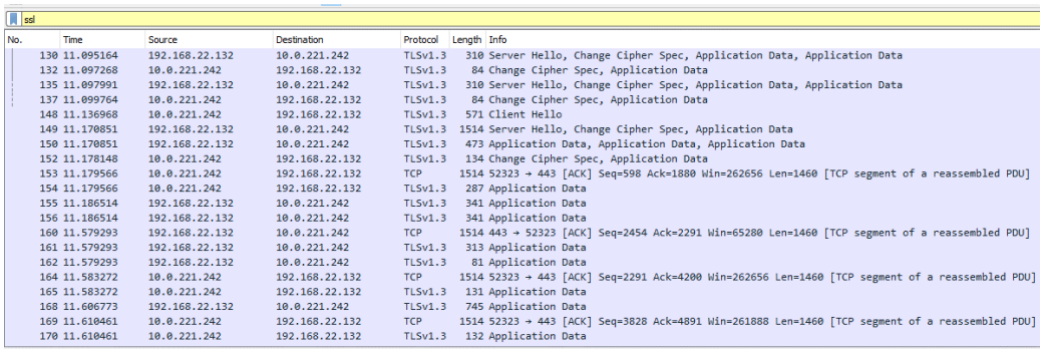
Informasi yang diperoleh sama dengan informasi sebelumnya yang membedakan hanya prosesnya yaitu *DELETE* Topeng, terdapat informasi detail yang lainnya *request method*: *DELETE*, *request URL*: /didik_nini_thowok/public/topeng/1, *request version*: HTTP/1.1, *Host*: 192.168.22.132\r\n.

3.3.2 Hasil Sesudah Implementasi SSL

Hasil pengamatan dan analisa dengan *Wireshark* dengan *capture* Sistem Inventaris setelah SSL selesai dipasang.

a. Percobaan Login Sistem Inventaris

Hasil *capture Ethernet*, muncul paket data yang tertangkap oleh *Wireshark* yaitu paket data *login* pada Sistem Inventaris dengan *filter* SSL. Daftar paket data yang tertangkap dilengkapi dengan nomor paket, *Time*, *IP Address* asal, *IP Address* tujuan, *protocol* yang digunakan yaitu TLSv1.3, serta panjang paket tersebut berapa *bytes*. Informasi detail data *Frame* 132 dengan panjang 84 *bytes*. Tanggal paket data berlangsung pada 14 September 2022 jam 12:35:15 dengan *protocol* TSLv1.3 dan menggunakan *port* 443. Informasi detail *ethernet II* terdapat *MAC* dari *source* dan *MAC* dari *destination*, dimana *MAC* dari *source* yaitu `Micro-St_51:8e:66 (00:d8:61:51:8e:66)` dan *MAC* dari *destination* yaitu `Routerbo_a1:2b:26 (74:4d:28:a1:2b:26)`. Informasi detail data *Internet Protocol Version 4* terdapat *IP Address* asal dan *IP Address* tujuan, dimana *IP Address* asal yaitu 10.0.221.242 dan *IP Address* tujuan 192.168.22.132. Paket data dapat dilihat pada Gambar 3.15

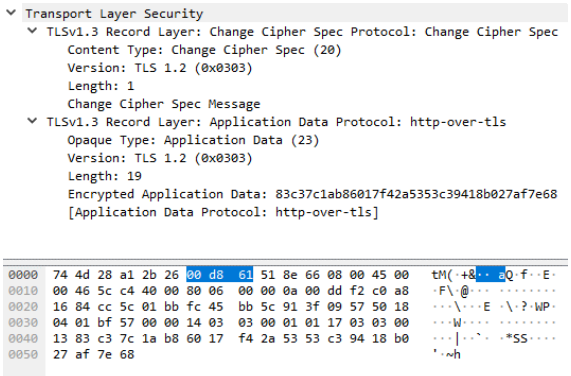


ssl

No.	Time	Source	Destination	Protocol	Length	Info
130	11.095164	192.168.22.132	10.0.221.242	TLSv1.3	310	Server Hello, Change Cipher Spec, Application Data, Application Data
132	11.097268	10.0.221.242	192.168.22.132	TLSv1.3	84	Change Cipher Spec, Application Data
135	11.097991	192.168.22.132	10.0.221.242	TLSv1.3	310	Server Hello, Change Cipher Spec, Application Data, Application Data
137	11.099764	10.0.221.242	192.168.22.132	TLSv1.3	84	Change Cipher Spec, Application Data
148	11.136968	10.0.221.242	192.168.22.132	TLSv1.3	571	Client Hello
149	11.170851	192.168.22.132	10.0.221.242	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
150	11.170851	192.168.22.132	10.0.221.242	TLSv1.3	473	Application Data, Application Data, Application Data
152	11.178148	10.0.221.242	192.168.22.132	TLSv1.3	134	Change Cipher Spec, Application Data
153	11.179566	10.0.221.242	192.168.22.132	TCP	1514	52323 → 443 [ACK] Seq=598 Ack=1880 Win=262656 Len=1460 [TCP segment of a reassembled PDU]
154	11.179566	10.0.221.242	192.168.22.132	TLSv1.3	287	Application Data
155	11.186514	192.168.22.132	10.0.221.242	TLSv1.3	341	Application Data
156	11.186514	192.168.22.132	10.0.221.242	TLSv1.3	341	Application Data
160	11.579293	192.168.22.132	10.0.221.242	TCP	1514	443 → 52323 [ACK] Seq=2454 Ack=2291 Win=65280 Len=1460 [TCP segment of a reassembled PDU]
161	11.579293	192.168.22.132	10.0.221.242	TLSv1.3	313	Application Data
162	11.579293	192.168.22.132	10.0.221.242	TLSv1.3	81	Application Data
164	11.583272	10.0.221.242	192.168.22.132	TCP	1514	52323 → 443 [ACK] Seq=2291 Ack=4200 Win=262656 Len=1460 [TCP segment of a reassembled PDU]
165	11.583272	10.0.221.242	192.168.22.132	TLSv1.3	131	Application Data
168	11.606773	192.168.22.132	10.0.221.242	TLSv1.3	745	Application Data
169	11.610461	10.0.221.242	192.168.22.132	TCP	1514	52323 → 443 [ACK] Seq=3828 Ack=4891 Win=261888 Len=1460 [TCP segment of a reassembled PDU]
170	11.610461	10.0.221.242	192.168.22.132	TLSv1.3	132	Application Data

Gambar 3.15 *Capture Wireshark* (SSL)

SSL yang berhasil dipasang pada Sistem Inventaris, merubah *protocol* HTTP menjadi HTTPS. *Protocol* yang digunakan adalah *protocol* TLSv1.3, maka *Wireshark* tidak dapat menangkap detail paket data seperti informasi tentang *"_token"*, *e-mail* dan *password*, informasi tersebut tidak muncul dikarenakan paket data yang tertangkap oleh *Wireshark* telah terenkripsi, informasi detail *Transport Layer Security* dapat dilihat pada Gambar 3.16



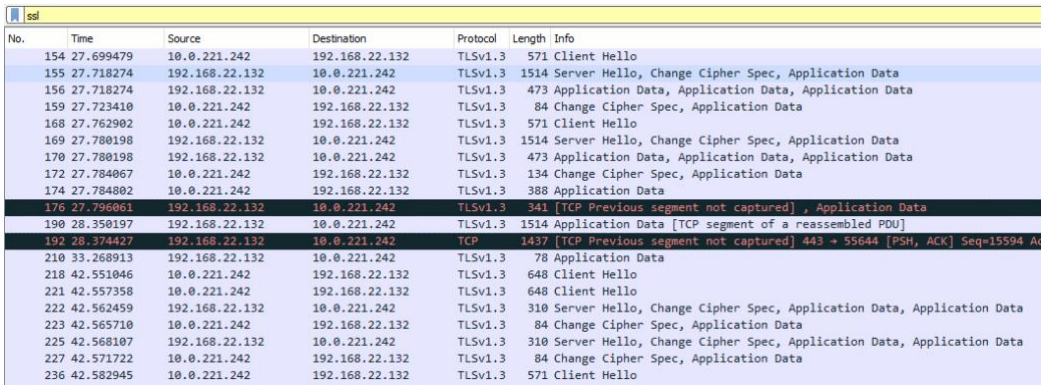
```
Transport Layer Security
  TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 19
      Encrypted Application Data: 83c37c1ab86017f42a5353c39418b027af7e68
      [Application Data Protocol: http-over-tls]
```

```
0000  74 4d 28 a1 2b 26 00 d8  61 51 8e 66 08 00 45 00   tM(·+&·· ·aQ·f··E·
0010  00 46 5c c4 40 00 80 06  00 00 0a 00 dd f2 c0 a8   ·F\·@··· ········
0020  16 84 cc 5c 01 bb fc 45  bb 5c 91 3f 09 57 50 18   ···\···E ·\·?·WP·
0030  04 01 bf 57 00 00 14 03  03 00 01 01 17 03 03 00   ···W···· ········
0040  13 83 c3 7c 1a b8 60 17  f4 2a 53 53 c3 94 18 b0   ···|··`· ·*SS····
0050  27 af 7e 68                                       '·~h
```

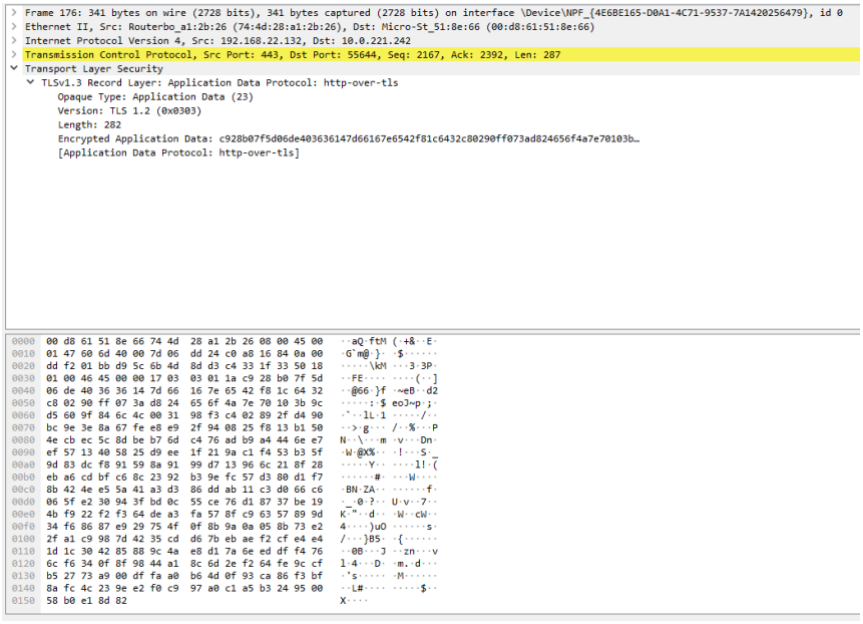
Gambar 3.16 *Transport Layer Security*

b. Percobaan Input Data dan Delete Data

Hasil *capture Ethernet* dengan *filter SSL* akan muncul paket data yang tertangkap oleh *Wireshark* yaitu paket data *input* data jenis topeng dan *delete* data topeng pada Sistem Inventaris. Daftar paket data yang tertangkap dilengkapi dengan nomor paket, *time*, *IP Address* asal, *IP Address* tujuan, *protocol* yang digunakan yaitu TLSv1.3, serta panjang paket tersebut berapa *bytes*. Tampilan detail informasi paket data yang tertangkap dapat dilihat pada Gambar 3.17 dan 3.18



Gambar 3.17 Percobaan *Input* dan *Delete* Data



Gambar 3.18 Percobaan *Input* dan *Delete* Data

4 Kesimpulan

SSL dipasang pada Sistem Inventaris yang sudah terpasang terlebih dahulu di dalam komputer *server*. Hasil pengamatan dan analisis dengan *Wireshark*, implementasi SSL pada Sistem Inventaris berjalan dengan baik dibuktikan dengan tidak adanya paket data berisi informasi penting dapat dilihat dan dibaca ketika dilakukan *sniffing*. SSL sangat berpengaruh terhadap keamanan data Sistem Inventaris karena SSL dapat mengenkripsi data pada saluran komunikasi antara *client* dan *server* sehingga dapat meminimalkan terjadinya serangan *illegal*, dibuktikan dengan adanya 3 kali percobaan oleh *user* yaitu proses *login*, *input* dan *delete*, data tersebut menjadi sulit untuk dibaca karena data telah 100% terenkripsi.

Daftar Pustaka

[1] I. D. Cahyani, "Sistem keamanan enkripsi secure shell (ssh) untuk keamanan data," *Jurnal Teknik Elektronika Fak Teknik Universitas Pandanaran*, pp. 1–8, 2011.

[2] D. N. Thowok, "Sanggar Tari Natya Lakshita," didikninikthowok.id.

[3] M. Christian, S. Suparni, and L. A. Utami, "SISTEM INFORMASI INVENTORY MENGGUNAKAN FRAMEWORK LARAVEL PADA CV. GRACE BHAKTI UTAMA BOGOR," *LRK*, vol. 1, no. 1, pp. 1–10, Jul. 2021, doi: 10.31294/larik.v1i1.456.

[4] H. Yuliansyah, "Perancangan Replikasi Basis Data Mysql Dengan Mekanisme Pengamanan Menggunakan Ssl Encryption," *Jurnal Informatika Ahmad Dahlan*, vol. 8, no. 1, p. 102982, 2014, doi: 10.12928/jifo.v8i1.a2081.

[5] R. Dastres and M. Soori, "Secure Socket Layer (SSL) in the Network and Web Security," *International Journal of Computer and Information Engineering*, vol. 14, no. 10, 2020.

[6] R. A. Setyawan, "Analisis Implementasi Load Balancing Dengan Metode Source Hash Scheduling Pada Procol SSL," *Jurnal EECCIS*, vol. 8, no. 2, p. pp.204-208, 2014.
[7] Sahren, "Implementasi Ssl Untuk Pencegahan Man in the Middle Attack Pada Ftp Server," *Journal of Science and Social Research*, vol. IV, no. 1, pp. 28–33, 2021.
[8] N. Novi and Z. Zaini, "Secure Socket Layer untuk Keamanan Data Rekam Medis Tumor Otak pada Health Information System," *Jurnal Nasional Teknik Elektro*, vol. 6, no. 3, p. 137, 2017, doi: 10.25077/jnte.v6n3.405.2017.
[9] D. Prayama, Yuhefizar, and Amelia Yolanda, "Protokol HTTPS, Apakah Benar-benar Aman?," *JACOST*, vol. 2, no. 1, pp. 7–11, Jun. 2021, doi: 10.52158/jacost.v2i1.118.
[10] H. Jammaluddin, "Analisis Keamanan Website Terhadap Sniffing Process pada Jaringan Nirkabel Menggunakan Aplikasi Wireshark (Studi Kasus: Simak Unismuh)," *Teknik Telekomunikasi*, vol. 1, no. 3, pp. 1–13, 2018.
[11] A. Hasibuan and E. Dalimunthe, "Implementasi Metode Client Server pada Penerapan Aplikasi Simulasi Ujian Akhir," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, p. 152, 2020, doi: 10.32493/informatika.v5i2.5614.
[12] D. Susianto and A. Rachmawati, "Implementasi dan Analisis Jaringan Menggunakan Wireshark, Cain and Abels, Network Minner (Studi Kasus: AMIK Dian Cipta Cendikia)," *Jurnal Cendikia*, vol. XVI, pp. 120–125, 2018.
[13] D. Hermanto and M. S. Anam, "Implementasi Sistem Keamanan Hotspot Jaringan Menggunakan Metode OpenSSL (Secure Socket Layer)," *Jurnal CoreIT*, vol. 6, no. 1, pp. 57–64, 2020.
[14] M. Haqqi and M. Badrul, "(021) 78839513 Fax.(021) 78839421 , 2 Sistem InformasiSTMIK Nusa Mandiri Jakarta," *Warung Jati Barat (Margasatwa) Jakarta Selatan.Telp*, vol. II, no. 2, p. 78839421, 2016.
[15] A. Tedyyana and R. Kurniati, "Membuat Web Server Menggunakan Dinamic Domain Name System Pada Ip Dinamis," *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, vol. 7, pp. 1–10, 2016.